18/151

Adjutant
General
School

Computer
Science
School

UNITED STATES ARMY
SOLDIER SUPPORT INSTITUTE

SPECIAL TEXT

ST 18-151

INCLUDES CHANGES 1 THRU 3

SYSTEMS ANALYSIS HANDBOOK

EDITION CODE 8 Edition Date: Nov 84

Prepared By
COMPUTER SCIENCE SCHOOL
UNITED STATES ARMY SOLDIER SUPPORT INSTITUTE
Fort Benjamin Harrison, Indiana 46216-5570

Community
Activities
School

Finance
School

Recruiting
and
Retention
School

ROTC
School

School
of
Music

FIT TO WIN
SOLDIER SUPPORT CENTER

Soldier
Physical
Fitness
School

TABLE OF CONTENTS

Page

CHAPTER 1. INTRODUCTION TO SYSTEMS ANALYSIS

SECTION 1. SYSTEMS OVERVIEW

1.1. Introduction 1-1
1.2. System Elements 1-2
1.3. Systems Analysis Defined. 1-2
1.4. Objective of Systems Analysis 1-2
1.5. Basic Elements in Sequence. 1-2
1.6. Summary 1-8

SECTION 2. SYSTEMS DEVELOPMENT

1.7. Introduction 1-10
1.8. The Project Team. 1-12
1.9. Phases of Systems Development 1-15
1.10. Phase I - Project Selection and Preliminary Survey. 1-16
1.11. Phase 2 - Problem Definition and Detailed Analysis 1-18
1.12. Phase 3 - Conceptual Systems Design 1-19
1.13. Phase 4 - Detailed Systems Design 1-20
1.14. Phase 5 - Programming and Procedures 1-22
1.15. Phase 6 - System Acceptance and Installation 1-23
1.16. Phase 7 - Systems Operation and Maintenance 1-24

SECTION 3. SYSTEMS ANALYSIS AND DESIGN

1.17. Introduction. 1-25
1.18. Information Processing Systems. 1-25
1.19. The Systems Analyst 1-26
1.20. Duties of the Systems Analyst 1-28
1.21. Role of the Systems Analyst in the Systems Development Process 1-30
1.22. Systems Analysis. 1-32
1.23. Overview of Tools and Techniques of Systems Analysis 1-33
1.24. Summary 1-37

CHAPTER 2. TOOLS AND TECHNIQUES OF SYSTEMS ANALYSIS

SECTION 1. INTERVIEWING TECHNIQUES

2.1. Introduction 2-1
2.2. Definitions 2-1
2.3. Use 2-1
2.4. Personnel 2-1
2.5. Top Management Interviewing 2-2
2.6. Middle Management Interviewing. 2-3
2.7. Operational Level Interviewing 2-3
2.8. Skilled Staff Interviewing 2-3
2.9. Planning and Interview 2-3

Page

2.10. Preparing for the Interview 2-4
2.11. Conducting the Interview 2-4
2.12. Tone of the Interview 2-5
2.13. Guiding the Interview 2-5
2.14. Ending the Interview 2-6
2.15. Following Up the Interview. 2-6

SECTION 2. DOCUMENT COLLECTION AND ANALYSIS

2.16. Introduction. 2-7
2.17. Existing Documents and Files 2-7
2.18. Factors of Data 2-7
2.19. Document Flow 2-8
2.20. Operations Performed 2-8
2.21. Document Format 2-8
2.22. Scope of the Study vs. Scope of the System 2-9
2.23. Determining Interface Requirements 2-10
2.24. How Data is Collected 2-11
2.25. Relationship: Document Collection and Interviewing 2-12
2.26. Document Collection Procedures 2-12
2.27. Document Classification 2-13
2.28. Developing a Document Classification Code 2-14
2.29. Code Application 2-15
2.30. Organization Coding 2-16
2.31. Instructions for Filling Out the DADS 2-19
2.32. Functions Performed by the DADS 2-23
2.33. Uses of the DADS 2-23

SECTION 3. MATRIX CHARTS

2.34. Introduction 2-25
2.35. The Grid Chart 2-25
2.36. Data Analysis Sheet 2-29

SECTION 4. FLOWCHARTING

2.37. Introduction & Definitions 2-34
2.38. The Systems Flowchart 2-37
2.39. Detailed Flowchart 2-41
2.40. Structure Charts 2-44
2.41. Data Flow Graphs 2-45
2.42. ANSI Flowcharting Symbols 2-48

SECTION 5. DECISION TABLES

2.43. Introduction 2-52
2.44. Parts of a Decision Table 2-52
2.45. Limited Entry 2-53
2.46. Example 2-53
2.47. Steps in Constructing a Decision Table. 2-55
2.48. Extended Entry 2-59
2.49. Mixed Entry 2-60
2.50. Summary 2-60

Page

SECTION 1. CONCEPTUAL SYSTEMS DESIGN
3-1. Introduction 3-1
3.2. User and System Requirements. 3-1
3.3. Meeting Requirements 3-1
3.4. Organizational Functions 3-2
3.5. Conceptual Systems Design Package 3-4
3.6. An Approach to Conceptual Systems Design 3-7
3.7. Conceptual Design and the Systems Analysis Tools. 3-8
3.8. Determining I/O DAS 3-9
3.9. Determining Files Specifications and Processing Rules 3-10
3.10. Decision Logic 3-12
3.11. Relationship Between Conceptual Design and Detailed Design 3-12

SECTION 2. INPUT/OUTPUT MEDIA
3.12. Output Media and Forms Design 3-13
3.13. Input Media and Forms Design 3-21

SECTION 3. MASTER FILES AND MEDIA
3.14. Introduction 3-29
3.15. Terminology 3-29
3.16. General File Characteristics. 3-29
3-17. Buffering 3-33
3-18. Magnetic Tape Master Files. 3-35
3-19. Direct Access Storage Concepts. 3-37
3-20. Basic Methods of Disk File Organization 3-38

SECTION 4. FLOWCHARTING THE NEW SYSTEM
3-21. Introduction. 3-44
3.22. Definition 3-44
3.23. Reference Material 3-44
3.24. Factors to be Considered During Flowchart Preparation 3-45
3.25. Symbols 3-50
3.26. Construction Guidelines 3-50

SECTION 5. SYSTEMS RELIABILITY
3.27. Introduction 3-60
3.28. The Areas of Internal Control 3-61
3.29. Input Controls 3-63
3.30. Output Controls 3-67
3.31. Hardware Controls 3-68
3.32. Programmed Controls 3-70
3.33. Organization Controls 3-73
3.34. Programming Controls. 3-75
3.35. Operating Controls 3-76
3.36. Summary 3-78

Page

SECTION 6. INTRODUCTION TO TELEPROCESSING SYSTEMS DESIGN
3.37. Introduction 3-78
3.38. Characteristics of Teleprocessing Systems 3-78
3.39. Management Defined Goals. 3-79
3.40. Analyst Defined Parameters. 3-80
3.41. Selection of Data Terminals 3-82
3.42. Basic Tasks of Terminals 3-89
3.43. Programmable Terminals 3-93
3.44. The Communication Facilities. 3-95
3.45. Network Design 3-97
3.46. Access Methods 3-103
3.47. Error Correction Schemes. 3-109
3.48. Net Data Throughput 3-114
3.49. Modems 3-116
3.50. Telecommunications Control Unit 3-121
3.51. Software 3-122
3.52. Methods of Optimizing 3-124
3.53. Summary 3-130

CHAPTER 4. SYSTEMS DOCUMENTATION

SECTION 1. GENERAL
4.1. References. 4-1
4.2. Purpose 4-1
4.3. Scope 4-1
4.4. Implementation 4-1
4.5. Objectives 4-2
4.6. Size of Document Types. 4-2

SECTION 2. DOCUMENT TYPES
4.7. Introduction 4-4
4.8. Functional Description. 4-4
4.9. Data Requirements Document. 4-5
4.10. System/Subsystem Specification. 4-6
4.11. Program Specification 4-7
4.12. Data Base Specification 4-8
4.13. Users Manual 4-9
4.14. Computer Operations Manual. 4-10
4.15. Program Maintenance Manual. 4-11
4.16. Test Plan 4-12
4.17. Test Analysis Report 4-13
4.18. Implementation Procedures 4-14

SECTION 3. PROJECT COMPLEXITY
4.18. Introduction. 4-41
4.19. Determining Complexity. 4-41

SECTION 4. APPLICABLE DA FORMS
4.20. Introduction 4-45
4.21. List of DA-Approved Forms 4-46
4.22. Forms Preparation Instructions 4-49

Page

SECTION 5. ADS SYSTEM AND COMPONENT IDENTIFICATION
4.23. Introduction 4-56
4.24. System Identification 4-56
4.25. ADS Systems Components Identification Codes 4-56

APPENDIX A. FORMS.

Logic Chart (DA Form 4735) A-2
Decision Table (DA Form 4736). A-3
File Specification DA Form 4737) A-4
Record Specification (DA Form 4738). A-5
Input-Process-Output (DA Form 4739). A-6
Input-Process-Output Cross Reference (DA Form 4740). A-7
Input-Process-Output Extended Description (DA Form 4741) . . A-8
Cycle Summary (DA Form 4742) A-9
Control Card Preparation (DA Form 4743). A-10
Control Card Preparation (Continuation) DA Form 4743-1). . . A-11
Run Setup - Console Operating Information (DA Form 4744) . . A-12
Run Messages and Responses (DA Form 4746). A-13
Output Reports (DA Form 4748) A-14
Reproduced Output Reports (DA Form 4749) A-15
PCM Processing Instructions (DA Form 4751) A-16
Recovery Instructions (DA Form 4751) A-17
Program Revision (DA Form 4752) A-18
Document Analysis and Data Sujmmary (FBH Form 32-0-2) Reverse Side A-20
Grid Chart (FBH Form 32-0-14). A-21
Data Analysis Summary (FBH Form 32-0-17) A-22
Terminal Screen Format (FBH Form 32-0-16). A-23
Form Layout (FBH Form 7027) A-24

GLOSSARY OF TERMS. G-1

TESTING MATRICES T-1

CHAPTER 1 -- INTRODUCTION TO SYSTEMS ANALYSIS

Section 1. SYSTEMS OVERVIEW

1.1 INTRODUCTION. Any discussion of systems development must begin with an understanding of the term system. In the very broadest sense, a system is defined as "an organized or complex whole; an assemblage or combination of things or parts forming a complex or unitary whole". The definition covers systems of varying complexities - from an ignition system in an automobile to the extremely complex system used to place astronauts on the moon. The following chapters are limited to a discussion of business/government systems - business organization systems and the information systems which receive input from the environment, and process this input in some way to produce an output. Concrete definitions of both types of systems are provided below:

a. Business/Government Organization System: A group of people who gather and process material and information resources to produce a product or service.

b. Information System: A group of people, set of procedures and data processing equipment which select, store, process, and retrieve data to provice information for managers to make timely, more accurate decisions.

A characteristic common to these systems is that they receive inputs from their environments and process these inputs into outputs which are exported back to the environment. Both types of systems receive inputs from society in the form of people, materials, money and information. These are transformed into outputs such as products, goods, and/or services. In a business system, for example, money and the commercial market provide a mechanism for the recycling of resources between the organization and its environment. All systems can be analyzed in this manner, using the input, process, and output perspective (Figure 1-1). This perspective is vital to understanding the analysis and design of information systems.

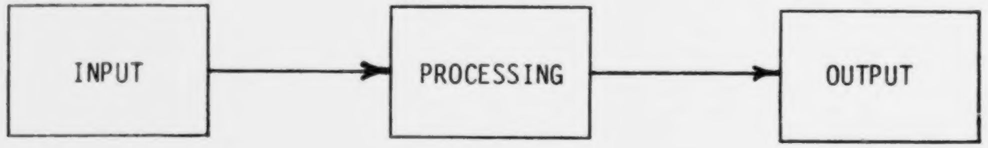


Figure 1-1. Model of a System (to include information systems).

1.2. SYSTEM ELEMENTS. We have defined a system as an organized complex whole designed to accomplish a particular function. It is important that we have a clear understanding of the elements that constitute a system.

A system is built on individual elements, or building blocks, each contributing to form an organized, integrated entity. The analyst who designs a new system must understand these elements and the relationship between them. An analyst cannot attempt to design a new or revised system unless he fully understands the existing one. If none exists, he must understand what is desired to be accomplished.

1.3. SYSTEMS ANALYSIS DEFINED. Systems analysis is the application of the scientific method to business or commercial problems. It is the separation of anything into its component parts, and the study and evaluation of these parts to see if there are better ways of meeting the needs of management.

1.4. OBJECTIVE OF SYSTEMS ANALYSIS. The objective of a systems analysis effort is to gain a realistic insight into a system and the related problem areas so that an improved system can be designed. Thus, the design effort cannot begin until the analysis is completed. This does not mean, however, that the analyst does not make some mental notes concerning the new system while he is in the process of analyzing the present one. This text will consider systems analysis as an independent phase or step whose ultimate goal is the total understanding of an existing system and its associated problem areas.

The systems that are the subject of analysis may be manual systems, where we may simply make improvements in the existing manual system, or our goal may be the conversion to data processing equipment. We may, on the other hand, have small data processing systems where the goal is conversion to a larger, more sophisticated, single system. The material presented here considers both categories.

1.5. BASIC ELEMENTS IN SEQUENCE. To facilitate the understanding of a system, we present it as having six basic elements. The diagram in figure 1-2 will serve as a foundation for this text. The systems analyst must collect data on a system at every management level within the organization top management, middle nanagement, and operating personnel as well. He will then have a general concept of the current system and how the system is used. A systems analysis begins with a detailed study of the six basic elements in a logical sequence. We will now discuss these six elements.

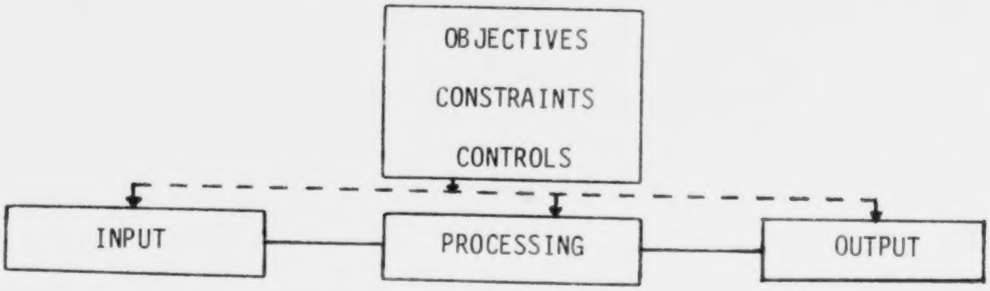


Figure 1-2. Elements of a System.

a. Objectives. The systems analyst must be aware of management's requirements for the system. In a very broad sense, he is also attempting to get management to define the problem as they see it. This explanation of the problem as defined by management aids the analyst in his attempt to determine if the problem, as defined, is the actual problem or merely the symptom(s) of an underlying problem. A systems study that is aimed at alleviating the symptoms of a problem will invariably be more complex, will be difficult to implement, and may not solve the real problem.

(1) The key to solving the real problem is to firmly establish the objectives of the system being analyzed. Once the objectives are clearly defined, the problems in meeting these objectives become clear, and an appropriate system for solving the problem(s) and meeting the objectives can be designed.

(2) In order to gain a thorough understanding of the systems objectives, we must define these objectives at several levels. First, there are the objectives of the total (information) system. The essential point about these objectives is that they contribute to the organization's mission. These objectives guide the overall design strategy in that they reflect the objectives of top management. The overall objectives should be broken down into more manageable sub-system objectives. In doing so, it is important to insure that -

(a) The sub-system objectives are consistent with the overall objectives.

(b) Whenever possible the sub-system objectives should be quantifiable.

(c) The objectives must be achievable.

(3) In a similar manner, the analyst must identify the major problems and break them down into manageable parts. Thus, the analyst is attempting to define the system's objectives/problems at each organizational level. In short, the analyst becomes aware of how the organization he is studying works and how the functions that are performed meet the stated management objectives. Often this requires formal training in a specific area. The experienced analyst can rely on his previous experience while newer analysts must learn about these business functions as they study the existing system.

b. Constraints. After studying the objectives of the present system, the analyst must determine any constraints. When the analyst knows specifically what the organization wants, he must then determine what limitations have been applied in the past and if these same or additional limitations will be applied to any future system. Typical constraints are as follows:

(1) Legal. Some information must be retained and specified reports prepared in a particular format for tax, regulatory, or audit purposes.

(2) Cost. Often, management will be willing to settle for a system that meets their minimal needs instead of one that is optimal but whose design is quite expensive. Cost is usually the most limiting constraint.

(3) Personnel. Management may dictate that the new system must not cause any major changes to the existing organizational structure or will not cause personnel reductions. On the other hand, management may want to save personnel costs, and, at the same time, restructure the organization. They may or may not be willing to retrain personnel for new tasks.

(4) Hardware. Often, analysts must study and design systems with the management directive that the systems so designed must be compatible to existing hardware or to hardware offered by a specific vendor. Also, lead time to obtain specific hardware might make it infeasible. This constraint can severely handicap the analyst by limiting the design alternatives open to him.

A study of the present system's constraints will provide the analyst with an insight as to why the existing system is designed the way it is and why the system operates as it does. Some of the current problem areas, will, no doubt, be traced directly to these constraints. When the evidence indicates that this is the case, the analyst should work with management to set more suitable limitations.

c. Output. After the objectives and constraints are understood, the analyst must determine what the system provides as output. Sometimes, the input required by managers and operating personnel of an organization is quite different from what the system really provides. This is a very fertile area for the identification of key systems problems.

As the analyst studies the existing outputs, he should begin developing ideas for their future revision. Based on his experience and the results of the analysis, the analyst must work with management in formulating new or revised report requirements.

When the analyst evaluates the current systems output, he must consider more than reports, since many systems have "saved" information in the form of updated files and other historical records or files that can be used for producing reports in the future. This "saved" information is often useful in providing answers to management regarding a particular system operation. A thorough knowledge of all system outputs, to include files, will provide for more effective use of data in the revised system.

In manual systems, the most common form of output is the printed page. The outputs for computer based systems may be printed forms, punched cards, magnetic tape, magnetic disk, cathode ray tube, microfiche, microfilm, and many other forms.

d. Processing. Once the analyst is aware of the outputs that are desired, he must recognize what data processing operations have to be performed on the data to produce the desired outputs. The basic operations which can be performed on data are: (1) capture; (2) classify; (3) arrange; (4) summarize; (5) calculate; (6) store; (7) retrieve; (8) reproduce; and (9) disseminate. All information systems entail some combination of operations. The particular operations that are required in a given system are determined during systems analysis as a result of identifying specific user information requirements. The logical or calculating algorithms which must be applied to the input data to produce the desired output must be thoroughly evaluated so that the goal of each operation is understood and is in its proper perspective. The analyst will ultimately be required to evaluate the processing performed to determine if the output is efficiently produced, or if some better method might be suggested.

e. Input. Now that the analyst understands the processing that is needed to produce the outputs of the current system, he must look at the source inputs that are used. Input is the information or raw data that will be fed into the system to serve as the basis for desired outputs. The input may consist of different media such as mark-sensed pages, optical character recognition (OCR) forms, punched cards, paper tapes, magnetic tapes or disks, when the present system is computerized.

The analyst must be thoroughly familiar with the characteristics, advantages, and disadvantages of each major type of input so he can incorporate the appropriate ones into his new design. Factors such as relative costs, processing speed, volume of data, storage capability, future expansion, and handling characteristics must also be considered. Listed below are a number of guidelines that should be considered in the design of input.

(1) Source data should be collected once, and only once, for input into the information system, regardless of the number of different ways it is to be processed.

(2) All source data should be thoroughly edited and validated as soon as possible after it has been collected.

(3) Data validated upon entry into the system should not be subject to subsequent validations of the same type in further processing.

(4) The accuracy of source data is directly related to the number of steps required to record, collect, and prepare the data for processing. The fewer the steps, the greater the degree of accuracy.

(5) Data should be stored in one place in the system data base, unless this is prohibited by hardware or software resource constraints.

f. Controls. After input, processing, and output have been fully examined, the analyst should see what types of controls are being employed to insure systems accuracy. Without good control procedures we run into the adage GIGO - garbage in, garbage out! When something goes wrong, it is quite common for people not knowledgeable in data processing to think that the computer has made the mistake. This is usually not the case.

(1) It is most often the fault of the systems analyst, the programmer, or a system user who has submitted invalid, incomplete, unverified data for processing. The stories of individuals receiving checks for such unreasonable figures as $500,000; or of customers receiving bills for as much as $7,500,000 are legendary. These errors are most often the result of programming or verification errors rather than machine failure.

(2) When analyzing the present system, the analyst should be looking for an entire system of controls and audits which will insure the accuracy of the total system. Procedures should be in effect to control the data from its initial entry through its dissemination. The following are examples of the types of controls the systems analyst should be looking for when analyzing a system.

(a) Input Controls. The object of input controls is to insure the accuracy and completeness of input. The responsibility for these controls is shared by the users of the system and the data processing personnel.

(b) Output Controls. These controls are designed to prevent the dissemination and use of erroneous output. Additionally, these controls include the capture, correction, and reprocessing of erroneous output. In short these controls insure the accuracy and completeness of output.

(c) Hardware Controls. These controls are designed to monitor and detect electrical failures, or the failure of moving parts. Hardware controls monitor the electro-mechanical reliability of the computer.

(d) Programmed Controls. Those elements of human judgement and division of duties once performed by humans form the basis of programmed controls. Programmed controls are controls that are coded as part of individual programs in addition to the instructions which process the data. These controls can be used in a variety of ways: (1) to detect erroneous input data; (2) to check the accuracy of programmed instructions; (3) to check compliance with various accounting policies and procedures.

(e) Organizational Controls. These controls are directed at the plan of organization to control the personnel within the data processing installation (DPI). Organizational controls usually take the form of procedures which spell out the duties and responsibilities of DPI personnel. These procedures serve to clarify the internal relationships of personnel within DPI, and the relationship between the DPI and the users of the system.

(f) Programming Controls or Standards. The objective of programming controls is to outline the standards, conventions, and procedures to be followed in the development, testing, maintenance, and documentation of programs and program changes.

(g) Operating Controls. Despite the accuracy of the data, the computer, and the programs, an error by the computer operator can negate all other control provisions. An operator error during machine set-up can introduce errors into a computer run. Operating controls and procedures attempt to reduce the possibility of such operators by providing explicit instructions to the operator.

(3) We have seen thus far that the analysis of a system involves the examination and study of six basic elements: objectives, constraints, outputs, processing, inputs, and controls. At this time, we will introduce one additional element that should be considered during analysis - feedback, and its relationship to the other six elements (figure 1-3).

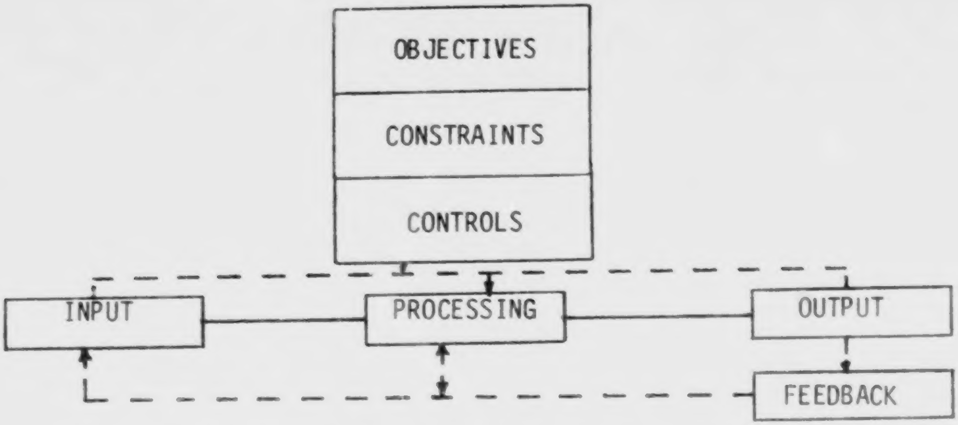


Figure 1-3. Relationship of the Systems Elements.

g. Feedback. This element is introduced here to give the systems analyst an idea of how a system must be designed to both support and facilitate future revisions. Change is a fact of life for the analyst and he must be aware that once a system is operational it must be constantly monitored to determine if some difficulty exists, or whether the outputs being produced are meeting the needs of the systems users at all levels. Through the process of feedback the system continually receives information from its environment that helps it to make the necessary adjustments. This feedback may come from any part of the systems' environment. It is wise to keep in mind Murphy's Law which states that anything conceivable which could go wrong almost certainly will go wrong. The analyst must be prepared for these contingencies. Therefore, he must build into his system a mechanism to obtain feedback, so that appropriate action can be taken to modify and improve the system when necessary.

1.6. SUMMARY. We observe that the analysis of a system requires a clear understanding of management objectives; the constraints imposed on the system; the relationships that exist between output, processing, and input; and the need for, and use of, controls and feedback.

Each of the elements of a system must be studied separately and then in relation to the entire system. During the course of the analysis effort, the systems analyst will discover basic problem areas and also develop ideas for a revised system. Only when the analysis of each element is completed will he begin to formalize these ideas.

A systems analyst need not have an extensive educational background in systems or business to be successful. What is required, however, is a logical mind, the ability to deal effectively with people, and a professional approach. The analyst must be able to sort through a voluminous amount of data, perceive problems as they really exist, and devise a better way of accomplishing the given objectives. When this has been accomplished, the role of the analyst switches to that of a salesman. He must "sell" management on the idea that he has, in fact, designed a better system.

Section 2. SYSTEMS DEVELOPMENT

1.7. INTRODUCTION. The systems development process is neither new nor unique. In a very basic form, everyone practices some type of system development as a part of his everyday life. As a youth, each person makes plans for educational and work experiences and implements them. Also, he plans for, reacts to, and works at the development of a satisfying, productive career. Later, as the head of a household, he continuously makes decisions. For example, he may want to buy a new color television set, but he cannot buy both a new refrigerator that he needs, and the television he wants. Therefore he must make a choice. The point is that the basic resources of time, money, materials, and skills are limited and must be applied in all human endeavors to achieve the most beneficial results for the efforts expended.

a. The same principles and techniques that an individual applies to his personal affairs apply equally to the development and implementation of information systems for business and government organizations. Information systems are more than the mere manipulation of numbers. These systems deal with actions, behavior, and interrelationships. Due to the obvious complexity of information systems, quantitative and mathematical skills are definitely required and are part of the picture, but they do not reflect the whole picture. The really vital skill in systems analysis, design, and development is the understanding of relationships. This skill in turn calls for basic reasoning and logic capabilities.

b. The cycle of problem identification and the development of an acceptable solution follows a logical pattern from the general to the specific. This is true of both small and large scale systems. The Polaris missile submarine is an example of a successful systems development project. From the standpoint of systems analysis and development, the Polaris missile program obviously moved from the general to the highly specific. The Polaris missile submarine was considered a total system. Application of logic and reason led to the subdivision of this overall project into a large number of separate specifically related subsystems. One group studied the necessary structure and internal components of the submarine mainframe. Another dealt with the missile itself. Others were involved with the missile guidance system. Literally tens of thousands of separate reasoning and logic efforts were build into the entire Polaris missile program.

c. The strategy or methodology used in the development of the Polaris missile submarine is an excellent example of a concept called the systems approach. This concept was first developed by the military as a means of controlling large research and development efforts that defied conventional management control techniques. The systems approach is the application of scientific methods to the study of any system. Emphasis is placed on the interrelationship of the parts or subsystems and how these relationships affect the performance of the overall system. The approach begins with a set of objectives, and it focuses on the entire design rather than on the design of components or subsystems. The collective action of the whole is all important in that the simultaneous action of the separate but interrelated parts produced a total effect greater than the sum taken independently. The results obtained by a team or a "system" of eleven football players is greater than that achieved by eleven individual players performing without integrated effort. A newly designed information system also will fall far short of optimum effectiveness unless it relates the various functions (subsystems) to each other and to the whole.

d. Now that the term systems approach has been defined, it is necessary to define the more basic term system. In an elementary way, a system may be described as a group of interrelated items acting together to accomplish a specific objective. This basic definition is illustrated by examples of some very diverse types of systems as shown below in Figure 1-4.

SYSTEM	ELEMENTS	BASIC OBJECTIVE
Factory	Men, Machines, Building, Materials	Production of Goods
Computer	Input Devices, Central Processing Unit, Output Devices	Processing of Data
Information System	People, Manuals, Procedures, ADPE	Timely Information

Figure 1-4. Examples of Systems.

e. The systems approach is particularly suited to the development of an information system because the design and implementation of an information system is managerially quite similar to the research and development projects for which the systems approach was originally developed. While these two types of development efforts would appear to be quite dissimilar, they have much in common. The common characteristics in all systems development projects allow them to be managed via the systems approach. The essential characteristics of a systems development project are as follows.

(1) Analysis of the problem as a single entity (system) in a particular environment.

(2) A definite set of goals and objectives.

(3) A definite beginning and ending.

(4) Use of scientific techniques and methods.

(5) Authority and responsibility vested in a single individual (project manager).

(6) Divisible into separate development segments or phases.

1.8. THE PROJECT TEAM. Because of the interrelated nature of all the separate things necessary for successful development and implementation, systems development projects have been characterized as team efforts. Although the people and groups involved have individual specialties, they must all be directed towards the accomplishment of specific, clearly defined objectives.

a. A problem of past information system development efforts was that, all too often, those initiating the development of information systems grossly misunderstood the nature of the development process. An unfortunate (and continuing) tendency on the part of user management is to regard information systems development as strictly a technical task. The result of this is a project team staffed primarily with junior and senior programmers and analysts. These people have been regarded solely as the caretakers of the computer; thus, undue emphasis has been placed on the computer, rather than on the problems of the organization as a whole.

b. A systems development project team should include specialized members from throughout the organization who are prepared to deal with information system development as a total organizational problem, and who can provide both guidance and evaluation as to whether the proposed solutions will actually work. These people are just as important, and possibly more so, to a systems development project as the programmers, software specialists, or even the hardware technicians. In today's highly computerized environment, persons with technical skills are somewhat abundant, but expertise on the specific problems of a given department or group within an organization must come from internal sources. This understanding of the organization's requirements and problems is one of the highly critical skills necessary to a successful system project team.

c. One of the key features of the systems approach is a team composed of specialists from many areas, often called an "interdisciplinary team". Depending upon the size and scope of the system to be developed, the typical specialists who could be on a project team are described below.

(1) Operations Researchers are specialists in the quantitative techniques which apply modern mathematical formulae to the analysis, simulation, and projection of results for systems alternatives. These specialists can assist the team in understanding the requirements for data needed in the design and development of information systems.

(2) Auditors must be involved whenever there are financial implications for a system under development. These specialists operate at different levels. In some cases, auditors determine the reliability of proposed new financial systems. In others, they must validate techniques, approaches, and results. Auditors play a key role in establishing internal controls, which are one of the basic elements of modern information systems.

(3) Human Factors Engineers are necessary to establish optimal relationships among the people, machines, and procedures that a system incorporates. They are specialists in the design of mechanical or control devices requiring human surveillance or operation. In information systems design, these individuals assist in designing or selecting the most appropriate devices, machines, or work stations for people entering information into the system, or deciphering on-line output.

(4) Software Specialists are needed when a modern computer system is being considered. Their function is to interface job performance programs with the computer equipment that will be used. They select and/or design such software elements as operating systems, executive programs, and communication control programs.

(5) Data Base Designers perform several essential tasks. They establish the formats for information records within the system, determine which records are needed, match data and storage capabilitaies with file content and magnitude, and also determine the structures and interrelationships of data files.

(6) Computer Science Specialists assist in equipment requirements, hardware configuration, reliability, maintainability, and other such data processing operational criteria. These personnel should be consulted before the programming language is selected. This is very important since relationships between programming languages and equipment memory size, peripherals, and other specifications are becoming nore and more closely interrelated.

(7) Facilities Engineers must be a part of the team because modern systems present fairly exacting requirements in terms of structures, electronic communications installations, maintenance, and other operational facilities.

(8) Programmers play an extremely important role in the systems development process. It is the task of the programmer to take the programming specifications provided by systems analysts/designers and convert them into detailed flowcharts (logic charts). The programmer must then convert these logic charts into computer language, and write, test, and debug the resulting programs. The programmer is also responsible for preparing instructions (the run book) to guide operations peronnel during production runs.

(9) Users or Customers are the persons who have been identified with a problem and who will ultimately use the newly designed system. The users must be closely involved with the systems development project since it is their needs and requirements which drive the system development process. Ultimately, the measure of a successful system is how well it meets the requirements of the users.

(10) Applications or Requirements Specialists assist the design team in much the same way as do users. Applications specialists have experience in industries, operations, or areas very similar to those in which the proposed system is to be used. These personnel are needed because users of the existing system are not necessarily qualified to judge the value of more advanced techniques, particularly in the development phases, and also because user experience has frequently involved outdated equipment and procedures. Therefore, specialists who are familiar with the environment in which the system will function and also are qualified to understand more advanced concepts of the new system are necessary to the development effort.

(11) Methods and Procedures Specialists are needed to effectively design the clerical functions and procedures for the new system. These personnel normally develop the detailed task lists and procedures manuals, and often design the standard forms which are needed in support of the information system.

(12) Systems Analysts/Designers work with the users to define system requirements. These user requirements and specifications are then translated into a form, and into terms, which programmers and methods and procedures specialists understand. The end product of analyst/designer efforts is the programming specifications. These specifications, when presented to a programmer, are normally in the form of a package of instructions that provide detailed information about programming the system.

d. As a general rule, the size and scope of the systems development project determines the size of the project or study team and the number of specialties represented. Some projects are so straight-forward that the specialities described above are not needed. However, on large projects, almost all of the specialties will be needed.

e. Actual membership on the team will vary widely with the phase of the project in progress at a given time. Membership may even vary with specific activities within any given phase. For example, operations research personnel should be consulted early in the project so that they may indicate the types of data needed for quantitative analysis. Not until the data is collected, tested, and verified does the operations researcher need to resume participation in the development project. The same type of participation applies to other needed specialties. In a manner very similar to the operations researchers, the user members must be present on the team at the onset. They must be closely involved in the preliminary design efforts.

Then they need not be present until later, during the test, conversion, and operation of the new system. Unlike operations researchers and users, programmers are not needed on a full-time basis until design and program specifications are well-established.

f. Obviously, the most important requirement of a project team is teamwork. Each specialist selected for a team must be acceptable to the other specialists and capable of working with them.

1.9. PHASES OF SYSTEMS DEVELOPMENT. A systems development project is not an on-going process. The project begins with the identification of a problem or a desire for improvement and ends when the new system is operational. It is precisely because of these definite limits that systems development lends itself so well to the project management or systems approach. Projects, regardless of the specific type, have clearly defined beginnings and endings and are composed of a series of activity sequences that must be performed in a given order. The basic structure of all projects makes it somewhat easy to subdivide projects at specific points. Generally, the points at which projects are subdivided coincide with changes in the nature of the work being performed. The change in the nature of the work being performed usually involves a shifting of the types of activities being performed and the skills that are required. This is why the project management concept works so very well. In addition, the project subdivisions are logical points for management review, approval, and allocation of resources for continuation.

a. As mentioned earlier, the systems approach is nothing more than the apoplication of scientific methods to the study of any system. Although there are many individual variations, scientific problem solution is generally divided into five stages:

(1) Defining the problem.

(2) Analysis of the problem.

(3) Selecting alternative solutions.

(4) Testing of the selected solution.

(5) Implementation.

b. When the systems approach and project management techniques were first applied to information systems development, the "classic" scientific problem solving approach was used. However, this methodology became increasingly inadequate as information systems became more complex and took on an interdisciplinary nature thus requiring the expenditure of enormous amounts of valuable resources. The use of resources became so great that management review and analysis of the systems development project at identifiable checkpoints became mandatory.

c. As larger and more complex information systems were developed, experience showed that development projects could not be neatly subdivided into the same five stages used in scientific problem solving. Other factors that led to discontinuance of the classic scientific method for managing the development of information systems were the cyclical or recurring nature of the systems development process and the overlap between the stages. The conceptual (general) design and the detailed design stages of systems development are similar. Also, major differences exist in the depth to which activities must be carried out, and in the resource expenditures necessary for their completion. Many factors led to the development of phases or checkpoints in the systems development process, but the two principal reasons for the refinement of the scientific method were (1) the needs of management to insure that the progress is taking place, and (2) the need to control the exoenditure of additional resources.

d. There is considerable disagreement on the various segments or phases of the systems development process; however, there is also a great amount of similarity and very little difference between the various approaches. In many cases, different terms are used to describe the same segments or phases that constitute the systems development process.

e. The systems development process can be described as consisting of a number of different phases. The development cycle used in this text consists of the following seven phases:

(1) Phase 1 - Project Selection and Preliminary Survey.

(2) Phase 2 - Problem Definition and Detailed Analysis.

(3) Phase 3 - Conceptual System Design.

(4) Phase 4 - Detailed System Design.

(5) Phase 5 - Programming and Procedures.

(6) Phase 6 - Systems Acceptance and Installation.

(7) Phase 7 - Systems Operation and Maintenance.

1.10. PHASE 1 - PROJECT SELECTION AND PRELIMINARY SURVEY. This phase of the systems development project begins with a request from someone outside the data processing activity (DPA) to conduct an investigation of the user's functional area to solve a particular problem or exploit an opportunity. This request can vary from a request for a single management report to a request for the design of a complete system. Depending upon the installation, the request may be communicated directly to the Information Systems Officer (ISO) who then contacts the AMO.

a. The Automation Management Officer (AMO) is the individual at the installation or activity charged with the responsibility for all ADP staff functions. The AMO performs a number of vital functions at the installation or activity, but he is primarily a special staff officer and the commander's principal advisor on all ADP matters affecting the command. Sometimes he is also the chief of the DPA and has overall responsibility for MIS, ADPE, and ADP resources. The AMO is, therefore, the individual responsible for meeting functional user requirements for ADP support.

b. The Information Systems Officer is the individual within each functional staff element responsible for providing ADP support to that element. ISO's in most cases are individuals in the functional staff element who are knowledgeable in ADP and are, therefore, designated as point-of-contact for all ADP matters. The ISO is the principal advisor to the chief of the functional element and also assists other members of that staff element. With regard to systems development, The ISO serves as the point of contact with the installation AMO.

c. Once the functional user's request is forwarded to the AMO, either directly or via the ISO, the AMO normally directs that a preliminary survey or investigation be initiated. The preliminary survey is an in-depth study performed on a one-time basis to gather data about the functional area or staff element requesting ADP support and to determine the role of information processing in that element. The purpose of the preliminary survey is to accomplish the following:

(1) Define problem objectives and scope.

(2) Evaluate the organizational setting.

(3) Consider possible courses of action.

(4) Select an optimal course of action.

(5) Determine organizational effectiveness.

d. In order to accomplish the above, the ISO or an analyst appointed by the AMO will interview key personnel in the functional organization and gather selected documentation. The interviews will normally be with management level personnel who know current operating procedures and who are in a position to point out any problems in the existing system. At this time, the ISO or analyst will review the available documentation in order to understand the methods which are supposed to be used in the system.

e. At the conclusion of the preliminary survey, the AMO will submit a report to the management (commander and staff) specifying what has been found to be the problems within the existing system and recommending a course of action. If warranted, the AMO may recommend that a detailed analysis and investigation of the current system be initiated to develop recommendations to solve problems that were found to exist in the current system. The preliminary survey report contains some extremely important information about the functional element requesting ADP support and can be used as follows:

(1) To better understand the overall functional organization and the functional area activities affected by the proposed system.

(2) As a tool in evaluating the validity of user requests for more ADP support.

(3) To explain current needs, attitudes, and future trends.

(4) To identify the source of any problems or significant constraints.

(5) To aid in planning the approach for detailed investigation and analysis.

(6) To aid in formulating questions to be raised by interviewers during the detailed investigation and analysis.

(7) To provide comparative data for cost versus benefit analysis.

1.11. PHASE 2 - PROBLEM DEFINITION AND DETAILED ANALYSIS. The preliminary survey is an extremely important step in the process of converting the user's request for ADP support into an operational system. After approval by management for continued study of the systems project, and with the assistance and support of the AMO, a project team is formed to conduct the full-scale investigation and systems analysis. This is the "fact-finding" stage of the system development process. It is during this phase that the systems analysis tools and techniques are employed to gain a detailed understanding of how the present system functions. Through the use of the systems analysis tools and techniques, the analyst and the other members of the study team will record the facts obtained in a systematic fashion. The tools and techniques used are not only "fact-finding" tools, but "fact analysis" tools as well. The study team is, therefore, able to analyze and evaluate the existing system. As a result of this analysis a number of alternative proposals should be developed which will solve the problems found in the current system. The study team will normally make a recommendation as to which plan they feel is the best means of satisfying the user's request. This stage of systems development must deal with the feasibility aspect of systems work. Just what does feasibility aspects mean? As the term is used here, it means a determination on the part of the study team as to whether or not the expected benefits to be realized from a proposed new system are sufficient to justify the expenditure of resources necessary for the problem's solution. The feasibility aspects are often referred to as the three feasibility criteria: Technical, Operational, and Economic feasibility. It is precisely because of the feasibility aspect that the detailed study and analysis of the current system becomes known as a Feasibility Study. Properly executed, it becomes a critical analysis of the functional organization's needs, and a comparison of the cost versus benefit relationships of the various alternatives that will satisfy those needs. The recommendation or report presented to management at the completion of this phase can be used to answer the following types of questions and considerations:

a. Is the functional user's request valid, and is an automated data processing capability desirable?

b. Is the recommended approach technically, operationally, and economically feasible?

c. What functions, tasks, and actions in the functional user's area can be automated effectively?

d. How much and what level of automation can be applied?

1.12. PHASE 3 - CONCEPTUAL SYSTEMS DESIGN. With detailed knowledge of how the existing system operates, it is now possible to systematically design a new system or modify an existing system to meet the needs of the user.

a. The process of systems design is concerned with the development of specifications for a proposed new system or subsystem which achieves the requirements determined during the detailed analysis of the existing system. Systems design can be defined as the drawing, planning, sketching, or arranging of many separate elements into a viable, unified whole. The detailed analysis phase reveals what the system is doing and what it should be doing to meet user requirements; the systems design phase is concerned with how the proposed new system will meet these requirements. At this point, the study team should be able to specify what the system must do, how well the system must perform, and what the system must provide for future growth. All of this analysis is based on a detailed definition of the goals of the system; e.g., cost reduction, better customer service, improved and more timely information.

b. The design activity which takes place in this phase is at a broad systems design level, and results in the production of conceptual specifications which will be the "blueprints" used in the detailed systems design phase that follows. Conceptual design begins with the identification of the overall function being performed by the user. Next, this overall function is subdivided into subfunctions. The subfunctions should be further subdivided and defined to a level of detail which adequately supports general or conceptual systems design. While it is not necessary to carry this process through to the lowest level of detail, it is necessary to ensure that no significant processes are overlooked.

c. Once the functions and subfunctions are identified, they must be described in terms of the processes that occur, the inputs that are necessary to support these processes, and the resulting outputs. By describing the system in this manner it is possible to produce the conceptual design of the new system: the specification of input and output to the system along with the decision criteria and processing rules. Only the content, not the media, of the inputs and outputs is specified. The actual media will not be determined until the detailed design phase. It is also in this phase that the study team determines the organization of files and the devices to be used.

d. The final product of this phase is a conceptual systems design package which represents the "blueprint" of the new system. The conceptual design package contains

(1) Output Specifications.

(2) Input Specifications.

(3) File Specifications.

(4) Decision Logic Tables.

(5) Processing Rules.

1.13. PHASE 4 - DETAILED SYSTEMS DESIGN. The detailed design phase is concerned with converting the conceptual design into a form that can be processed by the computer. While the conceptual and detailed design activities have been shown as being separate and distinct activities, there is actually considerable interaction between the two phases. The design process (conceptual and detailed) is a repetitive process rather than a sequential process as is implied above.

a. The process of detailed systems design includes -

(1) Prescribing the processing and decision rules by which input is converted to output.

(2) Selecting a file organization and processing scheme.

(3) Designing files.

(4) Incorporating controls to insure processing reliability.

(5) The physical formatting of layouts for inputs and outputs.

b. The result of detailed systems design is programming specifications. Development of clear and concise program specifications is one of the more vital steps in the entire systems development process. Even if the system is the finest ever designed, it will never be properly implemented if the programmers do not write the proper programs. If the program specifications are not communicated clearly, or the programs are written without the benefit of all the information being known, program modifications will be the inevitable result. When program modifications start being introduced, the probability of errors, unreliability, and delays in implementation of the system increase. Therefore, it is extremely important that the analyst or study team take as much time and care as possible to insure that all information concerning the programs is accurately described to the programmers in the programming specifications. These programming specifications should include the following:

(1) A narrative description or general overview of the entire system.

(2) A flowchart of the new system showing all programs to be written, inputs and outputs involved, and the source and destination of each.

(3) A listing and brief explanation of the programs within the system. This should include information about both the preceding program and the program following, with an explanation of how the interfaces with these programs is established.

(4) The format of each of the input, output, and master files to be created or processed.

(5) A detailed description of the processing to take place in each program. Decision logic tables may need to be attached to indicate the actions to be taken under various conditions.

(6) A description of the ways and means used for control of systems functions.

1.14. PHASE 5 - PROGRAMMING AND PROCEDURES. This phase of the systems development process pertains to the writing and debugging of the computer programs and the development of clerical procedures for the operation of the system.

a. After receiving the program specification, the programmer spends considerable time in reviewing the specifications and developing the detailed program logic needed to perform the processing required in the program. This logic is documented by means of a program flowchart (logic chart) and represents the steps that are required to solve the problem. Once the detailed logic charts are developed the program is coded in the computer language that has been selected. After coding, the program is key punched and deskchecked; e.g., each card is inspected for errors.

b. After the program has been desk-checked, and any errors have been corrected, the program is ready for testing. Testing and debugging are important because without workable programs the system will never produce the output for which it was designed. Before implementing the system, the analyst and programmers must be able to state with certainty that the programs will work exactly as they were intended to, and that procedures have been developed to properly handle any errors or mistakes that may come to light.

c. To insure effective testing of all programs, definite procedures are followed. The entire process of testing and debugging can be summarized as follows:

(1) Testing and debugging individual programs (Unit Testing).

(2) Testing all major routines using live test data.

(3) After testing with live data, perform "link" or "string" testing; e.g., test those programs together which depend upon one another.

d. Once programming and testing have been completed, the important task of documenting the system must be accomplished. The creation and maintenance of documentation is a vital task in any system. Often there is a tendency to neglect this task with the result that many hours of effort are required to reconstruct the input, output, and processing operations performed by the various programs that make up the system. The final documentation for a new system should include, at a minimum;

(1) Program documentation.

(2) Operator Documentation.

(3) User Documentation.

e. The development of system documentation must be planned. As a part of this planning, it is necessary to determine how much documentation is needed. Insufficient documentation can result in inefficiencies in the maintenance and operation of the system. Excessive documentation places a burden on the programmers, is inefficient, and is difficult to maintain and use.

1.15. PHASE 6 - SYSTEM ACCEPTANCE AND INSTALLATION. After the systems testing is completed and the documentation prepared, the new system is ready for implementation. If this testing and documentation have been thorough and complete it should be relatively easy to make the smooth transition from the old system to the new one.

a. In order to implement the new system successfully, there are a number of activities that must be performed. These activities include the following:

(2) Computer system acceptance testing.

(3) Convert the existing system files.

(4) Determine the conversion method to be used (direct, partial, or parallel conversions.

(5) Perform the "change-over" to the new system.

(6) Provide management orientation.

(7) Perform a post-implementation evaluation of the new system.

b. Once the system has been installed and is in operation, there is a need for systematic followup to insure that the system is operating as it is intended to, and that the original objectives of the system are being met. Analysts and/or programmers should be available to assist operations and user personnel in understanding the new system and in the solving of any problems that surface during implementation.

1.16. PHASE 7 - SYSTEMS OPERATION AND MAINTENANCE. Once accepted and installed the system moves from the development phase to the operation phase of its life cycle. The control of the system is now transferred from the analysts and designers to the group of personnel who will actually operate the system. The systems analysts/designers, however, must be concerned with the operational success of the system; and it is necessary that they make provisions for the successful operation of the system during the analysis and design phases. A system should be designed and implemented with the idea of facilitating its production operations. This can be accomplished most effectively through the use of operational standards. Operational or performance standards play an important part in controlling the operational system because they serve to locate and identify problems, remove bottlenecks, and maintain qualitative and quantitative output.

a. There are various kinds of performance standards, but generally they fall into three functional categories: (1) systems analysis standards; (2) programming standards; and (3) Data Center operations standards. The above three categories of standards are not arbitrary, but rather reflect a logical and natural separation of duties within the operational DPI.

b. There are a number of benefits to be derived from standards in the above three areas. Some of the benefits of standardization are realized immediately while others are of a long term nature. An effective standardization program provides the following benefits to an operational information system.

(1) A firm control over resources (time, money, manpower, materials, and machines) by providing a system of checks and balances.

(2) A possible reduction in operating costs through more effective use of resources.

(3) A means of evaluating the effectiveness of the whole system by applying a realistic performance standard to the feedback cycle.

(4) Simplification of the training process for new personnel.

(5) An effective means of two-way communication with the user management and the personnel within the DPI charged with meeting their requirements.

(6) Quality and accuracy in operations by clearly defining what output and what results are desired.

(7) Reduction of dependence upon individuals in an operation or in an organization.

Section 3. SYSTEMS ANALYSIS AND DESIGN.

1.17. INTRODUCTION. The term system is used everywhere today. We hear of a system of government, the solar system, a state highway system, the respiratory system within our bodies, or the payroll system within a business. An organization is a system composed of men, money, materials, methods, and machines.

It is very important to understand the nature of systems and systems concepts to deal successfully with the techniques of systems analysis and design. The systems approach is nothing more than the application of scientific methods to the study of any system, with emphasis placed on the relationship of the parts or subsystems, and how these relationships affect the performance of the overall system. The application of the systems approach helps to unify the action of the various parts of the system and produce an effect greater than the sum of its individual parts.

1.18. INFORMATION PROCESSING SYSTEMS. All organizations, regardless of their size, purpose, and structure, have some type of system for the processing of information. Although the information processing may be manual, in most organizations today, automated information processing systems are the rule, rather than the exception.

a. The information processing system of an organization, like the organization itself, consists of a group of sybsystems, most of which are common to all organizations, business or military. The subsystems which comprise the total information processing system are as follows.

(1) Production/Operations.

(2) Inventory Control.

(3) Marketing.

(4) Finance.

(5) Personnel.

(6) Research and Development.

b. The subsystems identified above represent the total information system for processing data to assist the organization in successfully accomplishing its intended purpose. These subsystems are interrelated and a breakdown in any one of the subsystems will have an adverse effect upon the effectiveness of the organization.

c. As noted earlier, there are several subsystems that make up a total information system. These entities just identified as subsystems of the information system also correspond to the functional or operational elements which make up the organization. In a typical business organization, for example, each of the elements identified would be a department with its own department head or director. Each functional element or department has its own mission which contributes to the success of the organization as a whole. In order to perform their assigned tasks, the managers and operational personnel within each functional element have specific information needs and requirements. The data processing department of the organization has the very difficult task of meeting the information needs of each functional element.

d. The data processing department, in order to meet the information requirement of each functional element, must develop and maintain a series of computer programs for each organizational element. What this means is that there is a group or "system" of programs which support each function. There is, therefore, a system of programs in support of the production/operations function, the inventory function, the marketing function, etc.; and, as pointed out earlier, these subsystems together represent the organization's information processing system.

1.19. THE SYSTEMS ANALYST. The functions or activities necessary for the organization to achieve its intended purpose are a basic fact of business life. Specialization by function is an accepted business practice, originating with the noted management theorist and practitioner, Frederick W. Taylor.

a. It is obvious that prior to the widespread use of computers, there were people whose job it was to insure that the various organizational functions were being performed well and for the benefit of the total organization. The successful integration and coordination of these functions called for the development of elaborate methods and procedures. These procedures were written and developed by personnel called methods and procedures analysts, operations analysts, management analysts, etc.

b. In most large organizations, these same tasks are still being performed by systems or systems and procedures staffs, both large and small. The tasks performed by these personnel are concerned with making the most efficient use of organizational resources. Some of the tasks performed are of an industrial engineering nature; e.g., making maximum use of space and improving the efficiency of production operations. Other typical duties fall in the range of general administration and cover forms analysis and control, records management and the performance of general management surveys and studies.

c. While a great deal has been said about the overall organization and the data processing system which supports it, no mention has been made so far of the organizational element responsible for the information processing system. This is the data processing department. The data processing departments of most organizations will differ considerably as to structure or configuration, but the three principal functions performed within a data processing department are shown in figure 1-5.

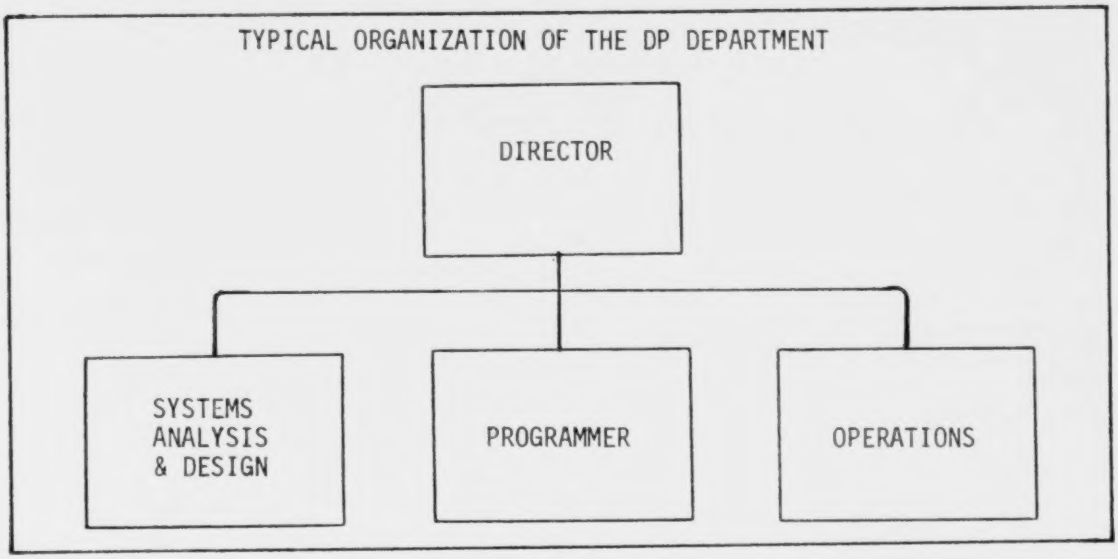

TYPICAL ORGANIZATION OF THE DP DEPARTMENT
DIRECTOR
SYSTEMS ANALYSIS & DESIGN
PROGRAMMER
OPERATIONS

Figure 1-5.

An explanation of the duties and tasks performed by the personnel in the data processing department were presented in the preceding section.

d. A system is a group of interrelated elements acting together to accomplish a specific objective. The organization, thus, has had "systems" analysts for a very long time - much longer, in fact, than it has had computers. These systems analysts (operations analysts, management analysts) usually comprise the systems and procedures staff whose duties are to analyze, design, and develop procedures for insuring the efficiency of functional operations.

e. The material presented in this text will apply to the types of analysts mentioned above. However, these tools and techniques will deal with the analysis, design, and implementation of data processing systems. Since most organizations are almost totally dependent upon the computer to satisfy a role of great importance. His increased importance lies in the fact that he is the catalyst and key individual in the analysis, design, and implementation of information systems.

1.20. DUTIES OF THE SYSTEMS ANALYST. The systems analyst is a member of the systems analysis and design element or group within the data processing department.

a. The position he occupies is responsible for the following duties:

(1) Assisting functional users in defining their information requirements.

(2) Determining if available data can be transformed into information.

(3) Evaluation of available technology for use in problem solution.

(4) Determining the best mixture of resources (Personnel, money, machines, materials, and procedures) for implementing the information system.

b. The systems analyst, whether in his normal duties within the data processing department or as a member of a study team, performs basically the same function. He is the person who is the interface between the users of the information system and the technical persons such as programmers, data base administrators, and operations personnel, who work within the system. This function is a logical as well as an all-important one because information systems are developed only to fill the information requirements of a wide variety of functional users throughout the organization. In this way, a systems analyst or a group of analysts insures that the system meets with the user's needs and desires and that the programming and operations deal with the more technical aspects.

c. In atttempting to define what attributes a successful systems analyst must possess, a number of different criteria can be used. These criteria will vary greatly from organization to organization, but there are criteria guidelines that can be developed and specified.

(1) Educational Background. The point being made here is the fact that an individual has had an extensive formal education is no guarantee that he will make a good systems analyst. Some type of college degree, in either business or computer science is desirable because it insures usually that the individual has been exposed to modern business concepts and the latest mathematical problem solving techniques.

(2) Experience. The systems analyst must have the ability to reason logically, and formal education is often one of the best means of sharpening these capabilities. But just as critical as logical reasoning is the need for the analyst to have experience in the practical application of

his skills. He should have demonstratd an understanding of systems concepts and a knowledge of the capabilities and limitations of automated data processing equipment (ADPE). While it is not necessary that an analyst be a "cracker-jack" programmer, it is important that he have a good knowledge of programming and data processing. The lack of knowledge of programming may result in the design of systems that are impractical from the standpoint of programming and implementation. In summary, an analyst should have a good background in systems concepts, business, and data processing; and this general knowledge should include a thorough familiarity with computer hardware and software capabilities.

(3) Personal Attribures. In addition to his educational background and experience, there are a number of personal attributes which are desirable for an analyst. These attributes may be summarized in the simple statement - he must be able to deal effectively and harmoniously with people.

(a) Success or failure for an analyst will often be determined by how well he deals with people, all types of people at all levels of the organization. In the course of solving a particular problem, it would not be at all uncommon for an analyst to interview top management personnel in the morning and a clerk at the operational level later in the day. Thus, one of the most important attributes for an analyst is to be able to converse with all types of individuals regardless of their level within the organization.

(b) In fulfilling his role as the point of interface betfween the functional users of the system and the persons within the data processing department, another key attribute is necessary: the ability to communicate. The analyst has the difficult task of communicating user requirements to the members of the data processing department; spoecifically programmers, operators, and other personnel who may be part of a study team for as particular project. Inevitably, conflicts will arise between the needs of users and the needs of data processing technicians, but the analyst must keep the channels of communiation open and diplomatically resolve all such conflicts.

(c) In addition to dealing with people, the analyst is concerned with designing the actual procedure that will constitute the system, or, if it is a complex system, assisting methods and procedures analysts in the development of proper procedures. He must, therefore, dra on his understanding of systems concepts and have the intelligence and ability to see the system as a whole. Most successful analysts are extremely resourceful and ingenious. They are inqusitive and seem to enjoy exploring alternative solutions to problems while keeping an open mind and remaining flexible until the time comes to determine the best way to solve a particular problem.

d. Once the fact-finding and fact analysis have been completed, and a new system has been "conceptualized", the analyst is ready to deal with one of the more difficult tasks. He must now communicate the mechanics of the proposed new system; the advantages and disadvantages of the design; the methods of programming and implementation; and, at the same time, deal with a number of considerations that will be brought up by management, programmers, and the actual users of the system. It is at this time that the ability to communicate clearly and concisely pays off since most problems which develop in data processing systems are the result of misunderstandings. These misunderstandings may include the analyst misunderstanding the needs and desires of users, or programmers and operators misunderstanding the analyst's directions for the processing that should take place in a given computer run. The ability to sell his own ideas and to communicate are among the key attributes possessed by the successful analyst.

1.21. ROLE OF THE ANALYST IN THE SYSTEMS DEVELOPMENT PROCESS. It is in the Problem Definition and Detailed Analysis Phase that the actual analysis of the current system takes place. This phase is often referred to simply as the systems analysis phase. What is systems analysis? It is an orderly and detailed procedure for collecting, organizing, and evaluating information about an organization with the objective of placing functions, documents, and data elements into relationship with each other so insight can be gained.

a. During this phase, the analyst assists users and operating personnel of the organization in identifying

(1) What information is required.

(2) When the information is required.

(3) How the information is required.

(4) For whom the information is required.

b. At the end of systems analysis, the analyst must be able to answer the questions -

(1) What takes place in the current system?

(2) What should the system be doing to meet user requirements?

c. Many problems are encountered during this phase, but the problem to which analysts must give special attention is that of getting users to quantify those requirements that are stated vaguely or qualitatively. Anothe extremely important contribution the analyst can make in this phase is in assisting the user in identifying what information is truly needed. This is often quite different from what users think they need, since their thoughts are based on some preconceived notion of what can be provided.

d. Systems design is the formulation and description of the nature and content of input, files, and output in order to show how they are converted b processing and procedures, and the development of a new or improved system. During the two design phases, conceptual systems design and detailed systems design, the analyst is given the opportunity to be creative and original in determining how the information needs of the user are to be satisfied. These phases require the analyst to use his knowledge of what technology is available, what techniques can be meaningfully employed, and what resources can be best utilized.

e. All of the analysis and design efforts come to a climax with the implementation of the new system. Implementation is not a distinct step, but involves two phases of the systems development process, Programming and Procedures and Systems Acceptance and Installation. The systems analyst must work with the other members of the project team to identify cutoff dates, to train and coordinate with user personnel, to instruct computer personnel, to install new procedures and forms, to test the new system, and to be on the alert for significant oversights or omissions from the early phases.

f. Once the system has been implemented, the analyst's job is not complete. In the last phase of the development process (Systems Operation an Maintenance), the analyst has an important role to play throughout the operational life of the system. He may be called upon to explain various aspects of the system to operating and management personnel. In addition, th analyst will have to assist in the identification of system malfunctions and the prescription of appropriate remedies. The analyst will be engaged in the continuous monitoring of the system, always seeking ways to improve its performance, as well as adding new elements as user's needs change. Finally, at some point in time, the analyst will again review the system in order to recommend a new, greatly improved information system or subsystem for the organization.

g. The role of the systems analyst is summarized graphically in Figure 1-6.

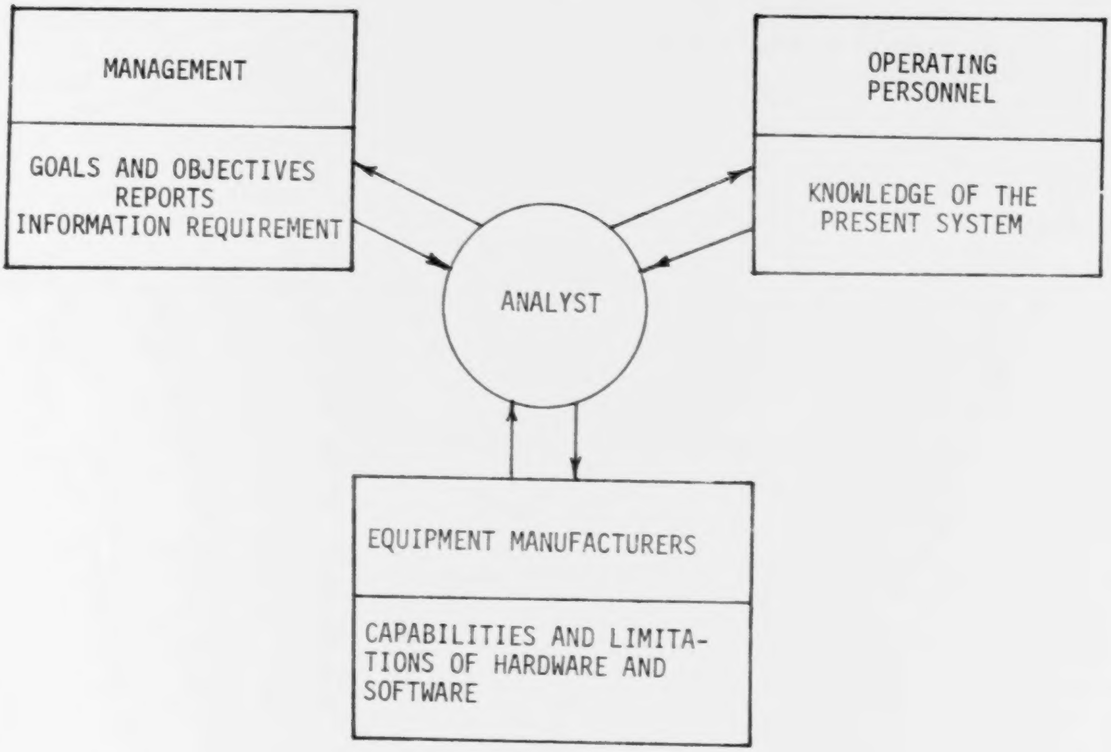

MANAGEMENT
GOALS AND OBJECTIVES
REPORTS
INFORMATION REQUIREMENT
OPERATING PERSONNEL
KNOWLEDGE OF THE PRESENT SYSTEM
ANALYST
EQUIPMENT MANUFACTURERS
CAPABILITIES AND LIMITATIONS OF HARDWARE AND SOFTWARE

Figure 1-6. ROLE OF THE SYSTEMS ANALYST

1.22. SYSTEMS ANALYSIS. All the information presented thus far in this chapter should point up the fact that the systems analyst is an individual who is skilled in the art and science of problem solving.

a. In the Problem Definition and Detailed Analysis phase, the systems analyst's job is to determine if there is a problem to be solved, and why such a problem exists. Once it has been determined that there is a problem, the analyst must determine what is taking place in the current system and also what the system should be doing if it is to meet user requirements. The act of answering the latter two questions is the "fact-finding" and "fact analysis" portion of the system development project.

b. In the detailed investigation the systems will use many of the same tools and techniques employed by the Information Systems Officer during the Project Selection and Preliminary Survey. The prime difference lies in the amount of investigation, and the depth to which the investigation is taken.

The basic fact-finding and fact analysis tools and techniques utilized in the detailed investigation and analysis are identified below, and shown as a matrix in figure 1-6 -

(1) The interview.

(2) The questionnaire.

(3) Document collection and analsis.

(4) Observing current operating procedures.

(5) Charting techniques.

(6) Sampling techniques.

(7) Tables/matrices.

(8) Models/simulation.

1.23. OVERVIEW OF TOOLS AND TECHNIQUES OF SYSTEMS ANALYSIS

a. The Interview. Within an organization, the interview is the most significant and productive fact-finding technique available to the systems analyst. The inverview is a face-to-face exchange of information; a purposeful conversation planned and controlled to gain a specific objective. The interview is the communication channel between the analyst and the organization, and the means by which the analyst determines what information is required and how these requirements can be met.

b. The Questionnaire. The questionnaire is merely a form containing a number of questions for which the analyst needs facts or opinions that may contribute to the success of the systems study. The use of the questionnaire should be limited to those situations where it is not possible to interview all the individuals associated with the procedures. The questionnaire can best be used as a fact-finding tool in the following cases:

(1) When the analyst and the recipient are physically separated and travel is prohibitive for both.

(2) When there are many potential recipients.

(3) When the information obtained will be used to verify similar information gathered from other sources.

	SYSTEMS ANALYSIS	Fact finding	Fact Analysis	SYSTEMS DESIGN	Forms/Reports	Data Base	Equipment Selection	Procedure Writing	Program Specifications	SYSTEMS IMPLEMENTATION	Personnel Training	Systems Training	Auditing	Maintenance	GENERAL	Communication	Problem Definition
TECHNIQUES																	
INTERVIEW		X				X	X				X		X	X		X	X
OBSERVATION		X									X		X	X		X	X
CHARTING		X	X		X	X	X	X	X		X		X	X		X	X
QUESTIONNAIRE		X					X					X	X	X			X
SAMPLING		X	X									X	X	X			X
DOCUMENT COLLECTION		X	X										X	X			X
TABLES/MATRICES		X	X			X	X	X	X		X	X	X	X		X	X
MODELS/SIMULATION		X	X		X	X	X		X		X	X	X	X		X	X

Figure 1- 7. Tools and Techniques of Systems Analysis

c. Document Collection and Analysis. During Project Selection and the Preliminary Survey, the organizational documentation describing structure and plans will provide an overview of policies, operating procedures, and documents that are supposed to be used in the system. In many organizations these documents may not be current (e.g., forms may no longer be in use and procedures may have been modified). It is important, therefore, that during Problem Definition and Detailed Analysis, the analyst obtain copies of these forms and, in addition, should obtain completed forms that have actually been used by functional personnel during normal activities. Samples of documents can normally be obtained when interviewing personnel associated with each procedure. Then through the use of the document analysis and data sheet (DADS), these forms will provide the basis for analysis of the existing procedures within the system.

d. Observing Current Operating Procedures. After the analyst has reviewed the current documentation, obtained copies of operating documents, and interviewed key personnel in an attempt to gather information concerning the detailed operation of the system and its problems, it is actually necessary to observe current operating practices. The purposes of this observation is multifold. It allows the analyst to determine what is being done, how it is being done, who does it, when it is done, how long it takes, where it is done, and why it is done. This is often a very revealing part of the analyst's effort for he is likely to discover that neither the systems documentation nor the statements from those interviewed truly reflect the system as it is in actual operation.

e. Charting Techniques. Charting is the technique of developing pictorial or graphic representations of some dimension of an organization or an organization's activities. Of the many tools and techniques used by systems personnel, charting is the one most closely identified with systems efforts. The charting technique enjoys wide usage because there aspects of the organization or its information processing activities which do not lend themselves to being represented by some type of chart. There are generally three types of charts: organization charts, physical layout charts, and flow charts.

(1) Organization charts provide information concerning reporting relationship, quantities of resources, interrelationships, and levels of authority and responsibility.

(2) Physical layout charts show the layout of the physical environment within the within the organization or the data processing department.

(3) Flowcharting is probably the most important of all the charting techniques used by the analyst. The flowcharting technique is one in which symbols are used to represent an activity, and show the interrelationships among elements in a system to varying levels of detail.

(a) Systems Flowcharts depict the system as a whole and show only subsystems or major elements.

(b) The flowchart of the present system is a Procedural Flowchart and is the graphic representation of a specific operation or of the data flow within an organization or system.

(c) The Logic Flowchart is the most detailed of all flowcharts and its function is to show the detailed logical or arithmetical steps involved in a computer program.

f. Sampling Techniques. Sampling is a statistical approach to arriving at an understanding of the present system. This technique is based on the theory of statistics and the laws of probability. Sampling techniques should be employed in those problem situations which would involve a tremendous amount of detail work of collecting and accumulating data to provide a given piece of information; or to operations which are unmeasurable. Sampling techniques also provide an effective means of making projections as to resource requirements. It is not unusual for an analyst to measure a certain activity on a limited basis and then project the resources required to perform this function for a full blown system.

g. Tables and Matrices. These are techniques used for representing varied and complex subject matter. These same techniques can be valuable to the analyst in a wide variety of situations, such as problem definition, analysis, procedures and program development, communications, and documentation.

(1) The Decision Table is an analytical tool which provides information in a tabular form that is precise yet easy to read and understand. Decision tables show the conditions affecting a given situation and the relationships that exist among the various conditions. The tables further indicate which action or actions are appropriate for a particular set of conditions. It is an excellent documentation tool and means for communication of processing logic.

(2) The Grid Chart is a tool used by the analyst to show the relationship among all of the documents collected. It only shows document-to-document relationships, however, and provides no information concerning the data elements in the system.

(3) The Data Analysis Sheet (DAS) is an analytical tool used to show the relationship between documents as well as between data elements used in the system.

h. Models and Simulation. A model is a representation. In order to produce information, the model must be a verbal or mathematical expression describing a set of relationships in a precise manner used to predict actions or events. Simulation is a technique for representing and experimenting with the essential elements of large systems by using the computer. At the heart of the simulation technique is the mathematical model which represents the real world system.

1.24. SUMMARY. As can be seen from the previous discussion, the duties and functions performed by a systems analyst are numerous. The types of systems that are designed and implemented by systems analysts include a wide variety of organizational functions. Regardless of the system being analyzed and/or designed, the analyst must perform his functions with discipline and by using well-defined methods if the system is to be successful. The chapters that follow detail many of the analysis and design techniques used by successful systems analysts.

CHAPTER 2 -- TOOLS AND TECHN
Section 1. INTERVI

2.1 INTRODUCTION. Systems analysis is
the scientific method to the solution of
any scientific research is to make a ger
"define the problem" and, at the same ti
relationships that bear on the problem
technique which allows the analyst to a
definition in a minimal amount of time.
possible to discover not only the facts
attitudes of the people working on the

a. As a data gathering techni
in that it provides a way of securing
of the persons being interviewed. The
party had previously realized may be d
discussion.

b. Properly conducted, the ir
to correct information; however, if im
include misinformation and can lead t
this extremely powerful fact-finding
masters effective interviewing techni

2.2 DEFINITION. An interview is a p
conducted to gain a specific objctive
information, and it is the communicat
members of the organization being st
as a discussion rather than a formal
that seems to be an eye-to-eye confr

2.3 USE. Interviewing is used to g
organization and the role of the inf
organization. In short, interviews
present system operates. It is als
required of the new system and how
can be used to dispel fear and to g
for new ideas or methods. Moreover
for analysts to establish rapport w
throughout the study.

2.4 PERSONNEL INTERVIEWED. Interv
staff within an organization. Sin
from the organization's top execut
must be flexible enough to deal co
situations. Consequently, the int
formal to somewhat casual.

QUES OF SYSTEMS ANALYSIS
WING TECHNIQUES

ften described as the application of
business problems. The first step in
ral examination of the situation to
e, gather facts and identify
nder study. Interviewing is a
complish the goal of problem
By conducting good interviews, it is
concerning the problem, but also the
roblem.

e, interviewing has a unique advantage
formation recorded only in the minds
significance of ideas which neither
veloped during the give and take of a

erview can be a short and direct route
roperly conducted, the results may
poor decision-making. The benefits of
ol can be realized only if the analyst
ues.

rposeful conversation planned and
It is a face-to-face exchange of
on channel between the analysts and
ied. An interview should be thought of
nterrogation session. The interview
ntation is doomed to failure.

ther data or information about an overall
rmation processing system within
re used to gain information about how the
used to gain information about what is
ese requirements can be met. Interviews
n support or understanding from the users
interviews offer an excellent opportunity
th user personnel that will be invaluable

ewing is conducted at all levels, line and
interviews may take place with everyone
e to mail clerks, the analyst's approach
ortably with all types of people and
rview procedures can vary from highly

a. The interview locations are also subject to wide variation, from the executive suite to the shop floor; thus, the success of an interview may well depend upon how well the analyst is able to adjust to these types of environmental changes. The adjustment process is further complicated by the qualities possessed by or lacking in the analyst. The qualities in question here are previous experience and education, and the ability to communicate effectively with personnel at the various organizational levels.

b. The titles used to describe the various organizational levels contacted during an investigation may vary from one organization to another. However, an individual's title conveys little meaning in itself, and must be evaluated in relation to the size and structure of the organization in which he works. Generally, we can consider four groups of personnel in any organization that will have to be interviewed, and the analyst must use the most suitable approach to each group. The groups of personnel that must be interviewed are as follows:

(1) Top Management -- Decides policies and overall direction.

(2) Middle Management -- Execution and interpretation of policies.

(3) Operations Level -- Performance of day-to-day operational activities.

(4) Skilled Staff (All Levels) -- Provides expert functional area guidance to other management levels.

2.5. TOP MANAGEMENT INTERVIEWING. The systems analyst contacts top management at the time the problem study is initiated. As mentioned earlier, the analyst relies heavily on interviews during the preliminary problem study. As a result of this initial survey, the analyst gains a broad understanding of the organization, and is told by top management of the goals, objectives, and constraints that will provide the direction for future study efforts. Following the initial interview, the analyst should obtain individual and collective interviews. At this level of the organization, he cannot merely adopt the role of questioner and expect to obtain all oF the information he needs. He must be as concerned with hearing opinions as he is in obtaining straight-forward answers. Top management's attitudes toward ADP and ADP systems will often determine the success or failure of a problem study.

The main points to be covered concerning interviews with top management are listed below.

a. Ascertain the future strategy of the organization. What are the long range goals of the organization and what specific strategies will be used to achieve these objectives?

b. Inquire about the organizational structure and current administrative procedures, particularly about the relative positions, duties, and responsibilities of the line managers.

c. Obtain ideas, opinions, and facts pertaining to organization objectives, major problem areas, and their relevance to the new system.

d. Avoid asking for trivial facts or precise details of methods or procedures. Save these questions for middle management and operating personnel.

e. Make every effort to provide top management with an understanding of data processing. However, avoid using "computer jargon". Insure that top management is aware that their enthusiastic support will be invaluable in obtaining willing cooperation from all levels of the organization.

2.6. MIDDLE MANAGEMENT INTERVIEWING. Middle managers are an important source of information for the analyst. Their importance to the analyst lies in the fact that it is the task of the middle manager to coordinate organizational resources (manpower, money, material, and information) toward the attainment of organizational objectives. This level of management is involved with the formulation of plans and procedures and the design of the operational systems that will result in the achievement of the objectives. The personnel at this level will be able to provide information relating to such areas as formulation of budgets, funds flow analysis, document flow analysis, personnel problems, and operational performance and/or constraints.

2.7. OPERATIONAL LEVEL INTERVIEWING. The people at this level are involved with the actual task performance in the organization. These are the individuals who perform the day-to-day activities and, for the most part, follow a routine pattern of work. In complex organizations, many of these individuals are professionals or highly skilled experts, as well as skilled and unskilled workers. At this level the analyst attempts to discover if the systems and procedures devised by the top and middle managers are actually being implemented in an efficient and effective manner.

2.8. SKILLED STAFF INTERVIEWING. The special staff that reports directly to various executives within an organization is also an excellent source of information. Due to the greater complexity within organizations, staffs have come to play a very important role providing service, information, and even control over organizational units. Since the special staff is such an integral part of organizational operations, it represents an extremely valuable source of information as the analyst seeks to understand the present system and design a better one.

2.9. PLANNING THE INTERVIEW. The interview is an extremely powerful tool available to the analyst when properly planned and well executed. A few points to remember when planning the interview are listed below:

a. Arrange appointments as far in advance as possible.

b. Prepare beforehand (and follow) an outline of topics to be covered. Don't become involved in day-to-day problems unless they impact on the study.

c. Obtain a letter of authority. A copy of this letter should always be carried with the analyst in case of need.

d. Question all levels of personnel about improvements they would recommend if they were in a higher position. If these recommendations are later adopted, give credit to the individual who made the recommendation.

e. Become familiar with the functional areas where personnel are to be interviewed. This can be done by studying materials collected during the preliminary survey.

f. Plan to interview personnel who represent both sides of significant topics. For example, the originators and the receivers of a given document should both be interviewed.

g. Learn to separate fact from opinion as soon as possible. Opinions of key personnel are extremely revealing and should not be ignored, but they must be identified for what they are.

2.10. PREPARING FOR THE INTERVIEW.

a. When making arrangements for the interview, give the manager an idea of its subject and purpose so that he will give some thought to the matter.

b. Decide specifically what is to be accomplished by the interview, what information you are trying to obtain from the interviewee, and the general approach you are going to take during the interview.

c. Identify the proposed interviewee's position within the organization. Also, learn his general job responsibilities, activities, etc.

d. Ask about the duties and responsibilities of each senior member of the manager's staff, and make discreet inquiries regarding their personalities so that any difficult individuals can be anticipated. Seek permission to interview the staff, explaining briefly the reason for this requirement.

2.11. CONDUCTING THE INTERVIEW. The most productive approach, both in leading up to, and during the interview, is the frank and forthright approach. The following points are essential.

a. Explain the objective of the problem study and the specific objectives of the interview. Give a brief description of the subject matter to be covered in the interview.

b. Emphasize the importance of both the interviewee and the interviewer to the success of the systems study.

2.12. TONE OF THE INTERVIEW.

a. Conduct the interview as informally as possible consistent with the need for organized, planned results.

b. Be courteous, but not apologetic. It is assumed that the interview is necessary and important, otherwise it would not be taking place. To make disparaging statements only serves to raise questions in the minds of the other parties and in no way adds to the interview.

c. The tone of the interview is a reflection of the mental and physical attitudes of both parties. If they are relaxed, interested, and informal, the interview will proceed smoothly. It is up to the analyst to create the mood and bring about this desirable situation. Render every possible assistance to the person being interviewed, but, in doing so, be genuine and sincere. Interest cannot successfully be pretended, it must really exist, since people are quick in noticing the presence or absence of genuine interest and they will usually respond in kind. If the analyst is relaxed both physically and mentally, and shows a genuine interest, the person being interviewed will reciprocate. Pay attention to what is said. Simply stated -- LISTEN!

2.13. GUIDING THE INTERVIEW. The interview should be limited to information gathering. While free discussion is encouraged, be careful not to be drawn off on too many tangents. In the event of digressions, gently guide the interview back to its intended course at the earliest opportunity. If such an opportunity does not naturally occur, the analyst should create such an opportunity. It is important that the analyst retain the initiative in guiding the interview and hold it to well-defined limits.

a. User cooperation is of prime importance and care should be exercised when attempting to resolve conflicting items of information or points of view. Therefore, the analyst must not appear to have any particular point of view neither must he agree.with statements which he believes to be incorrect.

b. During the course of the many interviews, the analyst may begin to form tentative ideas, but he must not commit himself to any specific recommendations or courses of action. The persons being interviewed may try to sound out the analyst's thoughts on particular situations, but under no circumstances should the analyst be drawn into any statements oF opinion. Care should also be exercised with respect to statements or comments dealing with tentative conclusions that may have been formed while interviewing personnel at the various organizational levels.

2.14. ENDING THE INTERVIEW.

a. Prolonged interviews are rarely desirable. If it does become necessary to discuss a work situation for more than an hour at one sitting, the discussion should be divided into two or more sessions, or, at least, puncuated by a break of some kind. Interviews are hard work for all parties concerned, but are more so for the person being interviewed because he experiences a certain amount of tension. This fact should be taken into consideration by keeping the interview as brief as possible. Unless due allowance is made, fatigue and the desire to "get it over with" may prompt the person being interviewed to make incomplete or inaccurate statements. As an interview progresses, it often becomes evidence that all of the selected topics cannot be covered in the allotted time, and the analyst may find it necessary to make another appointment at a time which is mutually convenient. No question that is important enough to have been included in the interview plan should be eliminated simply because time runs short on a particular day. While speed is desirable, it should never take priority over completeness.

b. Always end an interview in such a nanner as to permit informal checkbacks, if they are necessary, by introducing a comment that will leave the door open for subsequent interviews. Be sure to explain briefly that when flow charts and decision tables are completed for the procedure discussed in the interview, they will be brought back for verification. This brings the interviewee into the analysis process and contributes to making him feel that this is his survey as well. It also provides an opportunity for explaining the tools and techniques used in the systems study.

c. It is a good practice to end the interview by some physical act, such as pushing back a chair, standing, etc. The analyst should be alert at this time for significant statements oF information. Often, when the interviewee believes that the interview has ended, he may unconsciously relax any mental guards he has set up and then provide items of information that previously had been withheld.

2.15. FOLLOWING UP THE INTERVIEW.

a. Formally write up the findings from the interview.

b. Return a copy of the findings to the interviewee, asking that he provide verification, corrections, or additions.

c. File the interview results for future reference and analysis.

Section 2. DOCUMENT COLLECTION AND ANALYSIS

2.16. INTRODUCTION. While interviewing is the most important fact-finding technique, it is only one of the systems analysis techniques. Consideration must also be given to other sources of information that are available to the systems analyst. One very valuable source of information that is available to the analyst is the existing documents used and stored within the organization being studied. The documents found in most organizations can be classified as to those that describe how the organization is structured, those dealing with what the organization plans to do, and those that deal with what the organization is doing now.

We previously defined a system as an "assemblage or combination of things or parts forming a complex or unitary whole." The purpose of any system is the attainment of objectives according to a plan. The type of structure found in an organization simply reflects one manner in which the management of that organization has arranged the various parts or components to accomplish objectives. When those documents identified as describing organizational structure and plans are collected and studied, they will provide the analyst with a wealth of valuable information. While this data is valuable, it must be remembered that these documents may not reflect accurately what is currently happening. At best, they give the analyst an understanding of management thought, and the direction of the organization, at one point in time. The policy statements, procedures manuals, long and short range plans etc., when collected, may be completely out of date because it is not at all uncommon for structures, procedures, and plans to change while their documentation remains unchanged. The more current these documents are, the greater their value, since they can provide the analyst with an idea of how the organization should be functioning. In order to determine how the organization is actually functioning, those documents which describe what the organization does must be collected.

2.17. EXISTING DOCUMENTS AND FILES. The document collection and analysis techniques presented in this chapter will be oriented toward the systematic collection and analysis of those documents which describe what the organization does. This is not an easy task since the number of "operating" documents that are collected could number in the thousands. When collecting this volume of documents, it is necessary and practical to employ a worksheet for recording the information about a document as it becomes known to the analyst. This worksheet is called the Document Analysis and Data Sheet (DADS).

2.18. FACTORS OF DATA. The best source of information on data format and content are the physical documents themselves. It is important to obtain at least two copies (original and file copy) of every document produced or processed by the system. The documents produced and processed by a Data

Processing System give the analyst insight into how the current system operates. Very little activity takes place in an organization that is not caused by a document, or takes place without a document being used to record the fact that a particular event has occurred or an operation has been performed. In establishing how the current system operates, the analyst is primarily concerned with four factors of data.

a. Document Flow.

b. Operations Performed.

c. Document Format.

d. Equipment and Organization Utilized.

2.19. DOCUMENT FLOW. As the term is used here, it means the transmission of data from one individual or process to another. The flow may be between the processes in an automated system, or it may be between persons or offices on hard-copy documents, or it may be by means of electrical transmissions between locations remote from each other.

2.20. OPERATIONS PERFORMED. As used here, the term includes anything that is done to, or with, the system. These operations may include preparation of a report; recording information on a form; keying data onto punched cards, magnetic tape, or disk; or the storage of a document in a file. Operations performed also include calculations or other manipulations that transform the data; or that combine or merge two sets of data into a single file, or conversely, sort the data on one file onto several others. These last types of operations may be either automated, or done manually. In addition, decisions, analysis, and evaluations are other types of operations that are of importance.

2.21. DOCUMENT FORMAT. A "document" is normally thought of as being printed or written on paper -- "hard copy" as it is called normally. In automated systems, however, information may be transferred from one set of processes to another without ever being printed out in hard copy form. It is convenient to consider these transmittals also as documents whether on tape reels, disk packs, punched cards, or other modes. By doing so, we insure the complete coverage of data flow throughout the system.

Again, for the sake of convenience and completeness, oral transmissions of data from one person to another should be considered as documents. Tt is necessary those types of information notifications that serve to make inadequate procedures workable. If the oral transmission is formalized and repeated in the same sequence in each transmittal, it may be possible in the design of the new system to substitute an automated transmittal. Because of this possibility, it is also useful to record this type of information on a DADS.

2.22. SCOPE OF THE STUDY VS THE SCOPE OF THE SYSTEM. Earlier, it was pointed out that "document collection and analysis" consists of collecting and analyzing organizational documentation. It should be kept in mind, however, that all the documents within an organization may or may not be collected. The documents which are collected depend upon the size and scope of the systems study.

a. What is meant by the term "scope of the study" and who tells the analyst what the scope of his study will be? At the time the systems study is initiated, user management defines both the objectives and scope. Objectives provide a clear and precise statement of what is to be accomplished, while scope identifies the operation, subsystem, or functional area that will be affected by the study. More precisely, the "scope of the study" refers to the highest and lowest organizational elements, as well as to the lateral boundaries of the study. It is possible for the scope of the study to encompass all, or any, of the following items.

(1) The entire data processing system of an organization.

(2) Several subsystems within an organization.

(3) A single subsystem or functional area.

(4) A specific operation or activity within a subsystem or functional area.

b. The detailed explanation of the scope of the study by management establishes very definite boundaries within which the analyst must work. This makes it easier for the analyst to identify those individuals who must be interviewed and the organizational elements that must be contacted during the study. In short, defining the scope of the study permits the analyst to start visualizing the functional area, operation, or subsystem being studied as a "system". While it is imperative that the analyst think of the organizational work element being studied as a total system, he must also keep in mind the fact that his system may be but one of several subsystems within the organization, all of which contribute to mission accomplishment. Successful mission accomplishment for the total organization is made possible only from the contributions made by, and the interaction among, the various subsystems. Thus, if the system under study is to contribute to the successful accomplishment of the organization's mission, it must interact with other elements outside the scope of the study. This interaction may be with other subsystems within the organization, or it may be with activities or agencies external to the organization.

c. It now becomes obvious that the analyst cannot limit his study efforts to the organizational element being studied, but he must also expand his study to include elements in the external environment. Successful performance of any system is possible only through the interaction with its environment. This reality forces the analyst to expand the boundaries of the "system" to include the vital external elements. This gives rise to a new term, "Scope of the System." The "Scope of the System" refers to the total range of organizational elements, both internal and external to the organization, that must be studied in order to accomplish management objectives. The difference between scope of the study and scope of the system can probably be more simply stated in the following manner:

(1) Scope of the Study -- the Problem as defined by management.

(2) Scope of the system -- management's Problem as defined by the systems analyst or study team.

2.23. DETERMINING INTERFACE REQUIREMENTS. An understanding of the scope of the system is helpful in identifying interface requirements. What is an interface? What use is it in a system? Webster's Collegiate Dictionary defines an interface as "the place at which independent systems meet and act upon or communicate with each other." An interface can also be thought of as a confrontation between the source of, and the user of information. This may be in the form of a meeting between people, a written report, or it could involve communication between two different processing steps or moves of a computer system where, for example, a magnetic tape file takes the accumulated results of one processing step and passes it on to the next step. In coming to grips with the problem of interface requirements, the job of the analyst is to determine and document all data required as inputs to the system and as outputs from the system, whether currently in writing or on a form.

a. Interface requirements can be divided into three categories:

(1) Interfaces required within the organizational element (subsystem) being studied.

(2) Interfaces required within the parent organization but external to the organizational element being studied.

(3) Interfaces required with agencies or activities external to the parent organization in which the study is being conducted.

b. It can be seen that one of the primary tasks of the analyst is to identify interface requirements within the organizational element being studied. But, stated in other terms, the analyst is simply using another perspective in seeking to learn what information is needed within the scope of the study to insure mission accomplishment. This includes, of course, the types of information required from both operational and management points of view. The purpose, frequency, and other key information elements involved in the interface must be recorded, as must the identity of the units or subunits involved in the interface. These interfaces may be in the form of information submitted or reported via phone calls, telephone messages, meetings, written reports, magnetic tapes, cards, or other media. If the interface medium is in the form of a document, a separate Document Analysis and Data Sheet (DADS) should be filled out for each document. The analyst must isolate and identify all the required information elements that must be exchanged within the scope of the study, regardless of the medium of exchange.

c. When the analyst determines interface requirements within the organizational element being studied he has, in fact, identified the information elements that must enter and be used within the scope of the study. But what about the scope of the system? The analyst will know what information elements are needed within the scope of the system when he determines --

(1) Interfaces required within the parent organization, but external to the element being studied.

(2) Interfaces required with agencies or activities external to the parent organization.

2.24. HOW DATA IS COLLECTED. The two basic ways of collecting data are the study and evaluation of the organizational documentation and interviewing and observing key managers within the organization. Data gathering begins when the written documentation found in the organization is collected and reviewed. The analyst can form an objective idea of how the system functions by examining policy files, procedures manuals, and the various documents and forms used within the system. However, interviews with key personnel are by far the most informative and accurate methods of obtaining data bout a system. The interview technique allows operations to be reviewed first-hand with the people who perform them. These personnel can shed light on the system, and point out its deficiencies and shortcomings far better than any written document. Another distinct advantage of interviewing is that it provides a way of securing information that is not recorded anywhere except in the minds of the people being interviewed. Ideas and data, the significance of which was not previously recognized by either party, may be developed during the give-and-take of a discussion.

2.25. RELATIONSHIP: DOCUMENT COLLECTING AND INTERVIEWING: Once the analyst has a fundamental understanding of the system through evaluation of its documentation, he learns the system in detail during interviews. But, before he talks with the members of the organization, the analyst must be adequately prepared for the interview. From his study of the documentation, he should have a very good idea of the role and function of each employee to be interviewed, and should know where this individual fits into the organization. With this preparation completed, the analyst will attempt to learn all the fine points about the system from the interview. What he has learned from the documentation is simply how the system, if it were operating properly, SHOULD function; what he hopes tu learn from the interview is how the system ACTUALLY functions.

a. A good place to begin the interview is with the interviewee's position in the organization and his responsibilities or mission. This interview should result in the detailed outline of the functions, tasks, and actions being performed by the organizational element. The analyst accomplishes this very important task by working with, and assisting the interviewee in visualizing his organizational element as performing a single function. During the course of the interview, this function is examined and broken into smaller subfunctions. All subfunctions, thus identified, should continue to be subdivided and refined to a level of detail that insures complete understanding of the work performed by the organizational element.

b. By starting with a single function, breaking this function into sub-functions, and then breaking the subfunctions down even further, the analyst progresses from a general functional description to greater and greater levels of detail. This process allows him to obtain very detailed descriptions of the various tasks performed by the organizational element. He must also get the managers being interviewed to describe the functions in terms of the processes that occur, and point out the necessary inputs and resulting outputs. It is at this time that the analyst obtains the necessary two copies of each document used by, and being described by the manager being interviewed.

2.26. DOCUMENT COLLECTION PROCEDURES. Once functions are identified and all supporting documents (inputs/outputs) collected, it is imperative that this information be recorded in a standard manner. But, at this point, a very important question must be asked. Does the analyst have all the documents that are to be found within the organizational element being studied? The best means of insuring that all documents have been collected is to compile a master list of required reports. This can be accomplished by referring first to AR 335-11, List of Recurring Reports, and then by contacting the Reports Control Section of the local Comptroller's Office. The analyst can extract a list of Department of the Army (DA) requirements for reports within the scope of the study. This list will not be complete, since it will only reflect DA requirements. The local requirements for reports can be obtained from the Reports Control Section, and integrated into the list.

a. This combined list (DA and local requirements) should be taken to the highest levels within the scope of the study with a request that the list be audited, and any necessary additions or deletions made. The resulting updated list should reflect all documents, both formal and informal, used within the scope of the study. Once this has been done, the analyst can begin interviewing with considerable assurance that all documents have been collected. During the course of the interviews, he begins at the top of the organizational element being studied and proceeds downward through the chain of command. As each person at each level is interviewed, he is asked to review the compiled listing of reports and records, and to verify or correct the listing if it is appropriate.

b. At each level, and each time a document is collected, the analyst interviews the clerk or other individual concerned to obtain all of the pertinent facts about the document (source, use, volume, retention, etc.). This line of questioning may uncover more local forms which must also be collected. As answers to the various questions are obtained, they should be written down on the worksheet called the Document Analysis and Data Sheet (DADS).

2.27. DOCUMENT CLASSIFICATION. The outcome of a successful systems study will probably be a newly designed and more complex data processing system. A vast number of documents will have been collected during the interviewing and document collection processes. These documents give the analyst insight into how the existing system operates. But this insight cannot be realized unless the analyst and/or the study team members have some standard and universally applied technique that will aid in the study of the documents. It is essential that every document collected be assigned a classification. This classification is based on the source of each document and its use within the system being studied. By classifying documents, we define their function within the system. The classification also gives the analyst clues about what questions he should ask about a document. The analyst may take a different line of questioning when discussing a final output than he would when discussing a source input document. When documents have been collected, they can be classified by type, applying the following defintions:

a. Source Document. Source documents are the original documents on which are recorded the details of transactions - a document on which data enters the scope of the system for the first time. In a computer system, because of the nature of the data contained on these documents, the content may have to be converted to a computer-readable medium on a daily basis. An example would be the creation of punched cards from hand-written or typed source documents (hard copy).

b. Basic Record (Master File). A group of related facts or fields of information treated as a unit - to include hard copy (paper) documents, listings, and forms, as well as magnetic tape or disk files is considered a basic record. This basic record (or file) is permanent in nature and is used in day-to-day administration and operation. By its nature, a basic record resides permanently within the data processing system and does not ever exit the scope of the system. For example, an automated inventory control system would have a file (tape or disk) on which is recorded data that is fixed or permanent in nature; such as item name and stock number, quantity on hand, reorder point, etc. This file would be the basic record for the system. A system can have a number of basic records.

c. Intermediate Record (file). Intermediate records are called by a number of names: worksheets, summary sheets, summary cards, intermediate files, and work files. An intermediate record is temporary in nature, and is used to transmit data from one processing step to another. Like basic records intermediate records are internal to the system and do not exit the scope of the system.

d. Final Outputs (Reports). Final outputs are documents which are necessary for the accomplishment of a specific operational task or which are required for decision-making, planning, and control purposes. Reports are not media dependent and may be hard copy, punched cards, presentations on cathode ray tubes, or even magnetic tapes or disks. Final output documents are the end result (product) of the data processing system and, as such, should not require further machine processing. Final outputs may be used by managers and operations personnel within the organizational element being studied; or they may be needed to satisfy interface requirements within the scope of the system.

2.28. DEVELOPING A DOCUMENT CLASSIFICATION CODE. To assist in distinguishing the four type of documents identified above, it is necessary to develop some type of classification code. The codes used could be either alphabetic or numeric; however, a numeric code is used here. The codes are as follows:

CODE	TYPE DOCUMENT
0	Source Input Document
1	Basic Record
2	Intermediate Record
3	Final Output

Document classification codes are assigned to a document in terms of its relationship to the system being studied and are based primarily upon the source and use of the documents within the scope of the study. The document classification codes are equally applicable to manual and automated systems.

Any system can be viewed from an input-processing-output perspective. Since the organizational element being studied is also a system, it, too, can be studied from this same perspective. Code 0 documents are the vehicles upon which data first enters the system. The code 1 and 2 documents are necessary to permit the data to be processed within the system; and code 3 documents are the final outputs from the system. These documents may be used by the managers and operating personnel within the scope of the study, or they may leave the scope of the study to provide the necessary interfaces, both within the parent organization and with activities external to the parent organization.

2.29. CODE APPLICATION. Document classification codes are applied to documents as they are collected. Documents may be classified as source input documents (code 0) if they bring data into the system for the first time, and if there is no back-up data within the system which can be used to reconstruct these documents if they are lost or destroyed. Final output (code 3) documents are the end products of the system and require no further processing. They are used by managers and operating personnel within the scope of the study, and to interface with external elements. Final output does not act as back-up to any other documents found within the scope of the study. The basic and intermediate record documents are those documents which are often processed on a daily basis and do not fall into either of the foregoing categories. Basic record (code 1) documents are records or files used for day-to-day administration and operation and contain information that is relatively more permanent than what is contained in other types of documents. Intermediate record (code 2) documents are records or files which are used for information of a more temporary nature. These records/files simply carry data from one process to another. The frequency with which the different documents are created and processed should also be considered when making the distinction between basic records and intermediate records.

The general statements above may not apply to all coding situations. A document must, therefore, be evaluated based on its functions and its use to determine its proper classification. Each document should have only one classification code. The exception to the rule that each document will have a single classification code is the TURNAROUND document. A turnaround document is produced by the computer as output, forwarded to a user who adds a limited amount of data, and is then returned to the system for further processing. Sometimes, just the return of the document is "recording" the fact that a particular event has taken place. The information produced by the computer is readable by both humans and machines; only the new data entered by the user needs to be encoded prior to being re-input to the system for any new processing. In terms of code application, the turnaround document as produced by the system would be coded as a final output (code 3). When new data is added by the user and the document re-enters the system, it should be classified as a code 0 document since it is now bringing new data into the system. In both cases, the remarks block on the DADS should be used to provide further clarification.

2.30. ORGANIZATION CODING. To facilitate orderly collection, analysis, and filing of documents, some type of coding scheme should be used to identify the various levels of the organization at which a document was collected. Very often a document will be found at more than one level of the organization, and will be performing a distinctly different function at each level. Each major element of the organization being studied should be assigned a separate code. As documentation or reports are collected within this organizational element, the organization code should be written on the DADS so as to tie the document to the element where the function is performed. This same code can be used on such things as interview forms, task lists, and flow charts to aid in cross-referencing data. This technique aids in developing a filing system so that the mass of data accumulated during the study can be managed and controlled. A suggested method of organization coding is provided below:

CODE	MEANING
00.	Major element
00.0	Division
00.00	Branch
00.000	Section
00.0000	Unit
00.00000	Subunit

This type of organizational coding scheme can be superimposed upon existing organization or manning charts and then compiled into a code book for ready reference by the analyst or other team members. Following is a sample code book sheet:

CODE	MEANING
11.0	Accounting Office (Chief)
11.1	Administrative Division
11.2	Accounting Division
11.21	Control Branch
11.22	Ledger and Reports Branch
11.221	Ledger Section
11.222	Reports Section
11.2221	Compilation Unit
11.23	Analysis and Reconciliation Branch
11.3	Disbursing Division
11.31	Cash Control Branch
11.311	Check Control Section
11.32	Audit and Scheduling Branch

DOCUMENT ANALYSIS & DATA SUMMARY

1 of 1 SHEETS

1. TYPE DOCUMENT 0 - SOURCE INPUT	**2. IDENTIFICATION** Stock Receipt Card	**3. NO. OF PAGES** 1	**4. ORG CODE** 12

5. FREQUENCY	**6. VOLUME**		**7. NO. COPIES REC'D**	**8. NO. COPIES FWD**
Daily	AVG 90	PK 100	1	0

9. SOURCE ORGANIZATION Supply Depot

10. DESTINATION ORGANIZATION Stock Records Branch, Supply Agency

11. USE OF THIS DOCUMENT

Used to update balances of inventory as a result of stocks being received at depot.

12. DISPOSITION	FORWARDED		SUSPENDED		FILED	X	DESTROYED

13. HOW PREPARED Manually	**14. MEDIA** Card	**15. TOTAL NO. OF DATA ITEMS** 10

16. DOCUMENTS ORIGINATING FROM THIS AND BEING UPDATED BY THIS DOCUMENT.

Inventory Card

17. SEQUENCE BY 1-Depot, 2-Stock Number	**18. ACCESS REQUIREMENT** As Required

19. RETENTION CHARACTERISTICS

Filed for 6 months

20. BACKUP DOCUMENTS THAT CAN BE USED TO RECONSTRUCT THIS DOCUMENT.

NONE

21. REMARKS

NONE

22. DATE 16 DEC 84	**23. PREPARED BY** R. F. JUMPTHRU

FRH FORM 32-0 2, NOV 83
(ATSG-CS)

REPLACES JUN 81 EDITION, WHICH IS OBSOLETE

Figure 2-1.

1. ITEM NO.	2 SUB NO.	3. DESCRIPTION	4. NO. OF CHAR MAX	AVG	5. ALPHA/NUMERIC A/AN/N	6. NO. OF DEC POSITIONS
1		Depot				
1	1	Depot Name	20	18	AN	
1	2	Depot Street Address	20	19	AN	
1	3	Depot City & State	20	18	AN	
2		Supply Depot Code	1	1	A	
3		Data Stock Receipt	7	7	AN	
4		Received By	20	15	AN	
5		Item Number	2	1	N	
6		Stock Number	11	11	N	
7		Item Description	20	15	AN	
8		Unit of Measure	2	2	A	
9		Quantity Received	6	4	N	
10		Vendor Name	20	15	AN	
7. TOTAL MAX CHARACTERS OF INFORMATION →			149			

FBH FORM 32-0-2

Figure 2-2.

2.31. INSTRUCTIONS FOR FILLING OUT THE DADS. The Document Analysis and Data Sheet (DADS) is shown in Figures 2-1 and 2-2. The following paragraphs will explain, section by section, how the DADS is to be completed. As the following paragraphs are read, refer to the appropriate section on the DADS.

a. The Front side of the DADS:

(1) Type Document: Each document must be identified by type; i.e., source document, basic record, intermediate record, or final output. After the proper classification has been determined, place an "X" next to the appropriate entry. The document used as an example in Figure 2-1 has been determined to be a source input, and is therefore coded as such.

(2) Identification: Enter the document name, and/or the form number or PCN. The example shows a Stock Receipt Card.

(3) Number of Pages: This entry pertains to hard copy documents only, and should show the number of pages in the document received. For all other media (cards, tape, disk, etc.) enter "N/A". Since the example has been identified as a card, N/A was entered here.

(4) Organization Code: Using any appropriate coding scheme, enter the code which identifies the level at which the document was collected. In the example, an organization code of 12 has been assigned.

(5) Frequency: The frequency of the document is shown in this block, based on the information shown below:

(a) For code 0 documents, enter how often the document is received.

(b) For code 1 documents, show how often file updates or maintenance occurs.

(c) For code 2 documents, enter the frequency at which the file or document is produced.

(d) For code 3 documents, show how often the final output is created.

(e) Entries which should be used in this block, and their abbreviations are as follows:

Daily (D)
Quarterly (Q)
Weekly (W)
Semiannually (S)
As Required (AR)
Monthly (M)
Annually (A)

(6) Volume: The volume of source documents or final outputs will have a great effect on the workload the system must handle. Principally, there are two factors involved on defining volume - maximum and average volumes. Any possibility for file expansion should be noted. File volumes should be shown as outlined below:

(a) Code 0 should have both the average and peak numbers of documents received.

(b) For code 1 and 2 documents, enter the number of records in the file.

(c) For code 3 documents, enter both the average and peak number of final outputs produced.

(7) No. Copies Rec'd: Self explanatory

(8) No. Copies Fwd: Enter the number of copies of the document that are forwarded from the organization which uses it. The example shows no copies being forwarded because the document is kept in the organization.

(9) Source: For all documents, enter the name of the organization or agency that actually created or submitted the document.

(10) Destination: Enter the destination of the file or document. If the document has more than one destination (multiple addresses), list them all.

(11) Use: Enter a specific statement of the use and purpose of the document.

(12) Disposition: Enter the final disposition of the document. If the document is placed in a suspense file prior to final disposition, show this fact also.

(13) How Prepared: Show the means by which the document is prepared (manually, Mark sense, computer, keypunch, etc.).

(14) Media: Enter the media on which the document is created (tape, card, disk, CRT, etc.).

(15) No. Items: Enter the number of separate data elements or fields appearing on the document or record. Do not mistake items and sub-items.

Employee Name	Employee Address	Employee City & State
	Employee	

In the above record, the data element "employee" is made up of three sub-items - employee name, employee address, and employee city and state. This is an illustration of a single data item which contains three sub-items.

(16) Documents Originating From This and Being Updated By This Document (DOFS): It is vital that the relationship between the documents within the study be determined; therefore, enter the name and/or form number of any document that is created, updated, or, in some way, directly altered by this document. This section of the DADS is extremely important since the information in the DOFS is beneficial for use in preparing flow charts and matrix charts, both of which will be discussed later in the text. There will be no entries in the DOFS for code 3 documents because they are final products and undergo no further machine processing. Be sure that all documents listed in the DOFS are directly affected by this document. The example in Figure 2-1 shows that only one document is updated by the Stock Receipt.

(17) Sequence: Show the primary and subordinate fields (keys) used to sequence the file.

(18) Access Requirement: Enter how often the information on the document or file is retrieved or used. It should be noted that, in many cases, both the access requirement and frequency will be the same. On the other hand, the entries may differ, particularly in the case of basic records. Some basic records may be updated weekly (frequency) even though the information is retrieved from them on a daily basis (access requirement).

(19) Retention Characteristics: Enter the length of time the file or document is being retained in the system. This is usually stated in periods of time, a number of versions, or a number of cycles, and can be either written out or abbreviated as indicated below:

FACTOR	ABBR	FACTOR	ABBR
Cycle	C	Version	V
Day	D	Week	W
Month	M	Year	Y

Using the abbreviations shown above, then, a file which has a retention period of 2 cycles would be shown as "C02"; one which is retained 6 months could be shown as "M06", and so on. In Figure 2-1, the retention has been learned to be 6 months.

(20) Back-up Data: This section of the DADS is called the Back-up Data Section (BDS). It is used to identify those documents that back up the document being studied. In this section, enter the names of all of the documents that can be used to reconstruct the document if it should be lost or destroyed. Entries will be made in this section only for Code 1, 2, or 3 documents. Because code 0 documents have no back-up data within the scope of the system due to their originating outside of the system, the entry for code 0 documents will always be "NONE" or "N/A".

(21) Remarks: Include here any additional information that the analyst feels will help in the analysis of the document.

(22) Date: Enter the date the DADS is prepared.

(23) Prepared By: Show the name of the analyst who prepared the DADS.

b. The Back Side of the DADS. The reverse of the DADS is used to show the data elements contained in the document being studied, and will contain a complete description of each data element found in the document.

(1) Item Number: Each data element in the document will be numbered sequentially, beginning with 1. The number assigned is entered in this column.

(2) Sub-Item Number: If a data element is composed of a number of logical sub-elements, the sub-item entry is used to define each of the logical sub-elements. The entry for Depot in Figure 2-2 is an example of the use of sub-items.

(3) Description: Enter the name and/or number of the data element (field) being identified. Where possible, the Standard Data Element name (AR 18-12 series) should be used.

(4) Number of Characters: This entry contains columns where both the maximum and average number of characters in a data element can be entered. As a minimum, where the average number of characters is not known or cannot be learned, enter the maximum number of characters.

(5) Class: Enter the class of data; either alphabetic (A), numeric (N), or alphanumeric (AN). If a numeric field on a tape or disk file is to be in packed format, enter NP.

(6) No. of Dec Positions: Enter the number of decimal positions required. This number is included in the number of characters previously discussed.

(7) Total Characters of Information: Enter the total derived from adding up the maximum number of characters for all of the data items on the document.

2-32. FUNCTIONS PERFORMED BY THE DADS. The Document Analysis and Data Sheet is one of the most valuable tools available to the systems analyst in studying the present system. It is a record of all the known information concerning each document found in the present system. The DADS performs vital functions for the systems analyst. Several key functions performed by the DADS are listed below.

a. Facilitates uniform analysis of collected documents.

b. Records both background and detailed information concerning each document (i.e., source, use, disposition, etc.).

c. Identifies each document with a specific organizational element by means of the organization code.

d. Aids in developing a pattern of document flow (this is made possible through the use of the "Documents Originating From" section and the "Back-up Data" section).

e. Identifies data elements and ties them to specific documents.

f. Provides a source of data for final documentation of a system.

2-33. USES OF THE DADS. Once a DADS has been completed for each document in the system being studied, the analyst is ready to determine how these documents are used in accomplishing the nission of the organizational element being studied. To gain this understanding, the analyst needs to know the relationship between the data elements that make up these documents. In subsequent sections of the text, additional systems analysis techniques are presented which assist the analyst in determining document-to-document data element-to-data element relationships. In addition, charting techniques are used to give a graphic representation of these relationships. The charting techniques used are flow charting and matrix charting.

a. The flowchart is a graphic representation of the flow of documents through the system, the realted actions, and the sequences in which these actions are performed. It defines clearly and concisely, in step-by-step fashion, all inputs and outputs and the necessary resources and oerations employed to connect one to another. A correctly prepared flowchart will show each document (or file) in the system, and the following information:

(1) Document Source.

(2) Document Media.

(3) Sequence of actions through which the document passes.

(4) Processing which occurs within the system.

(5) Disposition of all documents.

The information contained in the DADS about each document makes the task of flowchart preparation considerably easier. The "documents originating from" section (DOFS) and the "back-up data" section (BDS) are of especial value to the analyst in flowchart preparation.

Matrix charting is the term used to refer to two other analyst tools, the "Grid Chart" and the "Data Analysis Sheet" (DAS). The Grid Chart is a form that is used to show the relationship between the inputs and outputs in the system. It does not, however, provide any insight into the relationship between data elements in the system. This function is performed by the "Data Analysis Sheet" (DAS). The DAS is the tool that shows the manner in which all data elements in the system relate to each other. What makes the DAS even more important is the fact that it not only shows the data element-to-data element relationships, but it also ties each data element to the forms and files in which it is used. The DADS is the basis for the preparation of both of these forms.

Section 3. MATRIX CHARTS

2.34. INTRODUCTION. At this point in the study of the present system, the analyst has collected documents, conducted interviews, and recorded the facts about each document (Document Analysis and Data Summary). The next step deals with the verification process. The section dealing with the document collection process pointed out that two copies of each document were to be collected. How does the analyst know that this has been done? Various Personnel on the design team took part in the collection effort and completed the Data Analysis and Data Summary(DADS)for each document collected. But how does the analyst know that no mistakes or omissions were made? Two types of matrix charts will now be examined that will assist the analyst in the verification process; the grid chart and the data analysis summary. One will assist the analyst in the verification process and the validation process, and the other will be used to validate the data elements.

2.35. THE GRID CHART. The first type of matrix chart provides a document-to-document relationship. The Grid Chart (Figure 2-3) graphically shows all input documents which enter the system, and the outputs that result from their processing. The interaction between the inputs and outputs of the system are easily seen through the presence of an "X" at the point of intersection. The grid chart has been developed on the principle that each document will function as an input or an output, allowing the analyst to identify those inputs which are supporting some output, and those outputs which are supported by some input. If one considers only the first document shown as an input in Figure 2-3, and studies those intersections opposite this entry, one c an readily determine all of the output which are supported by this input.

a. Completing The Grid Chart. The DADS will be the primary source of information necessary for completing the grid chart. The top part of the grid chart only serves to identify the area being studied as defined in the scope of the study. Completing the body of the Grid Chart in order to complete a proper analysis requires a systematic approach as shown in Figure 2-4.

GRID CHART

1 of 1 SHEETS

1. MAJOR AREA: Stock Records Processing

2. SUB AREA: NA

3. ORGANIZATION: Supply Agency

6. Frequency →			AR	W	W	D	AR	D
	7. Type Document →		1	1	2	3	3	3
		5. End Products or Outputs (TYPE DOCUMENTS 1,2,3) / 4. Input Documents (TYPE DOCUMENTS 0,1,2)	Inventory File	Due Out	Backorder	Shipping Order	Stock Transfer	Purchase Order
D	0	Requisition	X	X	╲	X		
D	0	Stock Receipt						
AR	1	Inventory File			X			X
W	1	Due Out	X			X		
W	2	Backorder	X				X	

8. DATE: 16 DEC 84

9. PREPARED BY: R. F. JUMPTHUR

FBH FORM 32 0 14, NOV 83 (ATSG-DS)

REPLACES SEP 83 EDITION, WHICH IS OBSOLETE

Figure 2-3.

Steps in Completing the Grid Chart

Step 1 - Arrange the DADS in order by Document Type (0, 1, 2, 3).

Step 2 - List the Title, Frequency, and Document Type of all input documents (0, 1, 2) down the left side of the grid chart.

Step 3 - List the title, frequency, and document type of all the output documents (1, 2, 3) across the top of the grid chart.

Step 4 - For each document listed on the input side of the grid chart, refer to the Documents Originating From section (DOFS) of the corresponding DADS. Place a half X (\) in the intersecting square for each document listed in the DOFS, and shown across the top of the grid chart.

Step 5 - For each document listed on the output side of the grid chart, refer to the Backup Data Section (BDS) of the corresponding DADS. Place a half X (/) in the intersecting square for each document listed in the BDS and shown down the left side of the grid chart.

NOTE: The basic record (code 1), and intermediate file (code 2) must be listed on both the left side and across the top of the grid chart. These two document types must be treated as both input and output within the system.

Figure 2-4.

b. Functions of the Grid Chart. If it is properly completed, the grid chart will help the analyst evaluate the old system and design a new system. Its functions are shown below.

(1) Verifies the document code.

(2) Verifies the information recorded in the Documents Originating From, and the Backup Data sections of the DADS.

(3) Insures that all the documents within the scope of the study have been collected.

(4) Establishes document-to-document relationships by relating -

(a) Input to output.

(b) Output to input.

c. Analysis of the Completed Grid Chart. After the grid chart has been completed, it must be thoroughly analyzed. There are two checks involved in making this analysis.

(1) The first of these checks is to determine if any of the four grid-charting rules has been violated. The four rules are:

(a) Source Input documents (Code 0 can never appear as end products (final outputs).

(b) End products (final outputs) can never appear as inputs.

(c) Intermediate records (code 2) must appear as both inputs and outputs. If the document coded as an intermediate file does not appear as an input, it may be a final output, and should be coded as such. If the document does not appear as an end product, it may be a source input and should be coded as such. Call-back type interviews can be used to verify these facts.

(d) Basic records (code 1) must appear as both inputs and outputs.

(2) The second check is to determine if there are any inputs that do not support output documents; or if there are any outputs that are not supported by input documents. The foregoing conditions can be detected by the presence of a half X (\ or /). It should also be determined at this time if we have collected any documents that are outside the scope of the study. These can be identified by the absence of either an X or half X (\ or /).

2.36. DATA ANALYSIS SUMMARY. The grid chart reflected the relationship between input and output documents, validated information relating to documents as it was recorded on the DADS, and accounted for the documents collected within the scope of the study. To further evaluate the old system, and aid in the design of a new system that will satisfy the requirements of management, the analyst must be able to see not only the relationships among documents, but also the similarity among input and output data and areas for files consolidation. The Data Analysis Summary, DAS (Figure 2-5) portrays this information graphically by providing a data element to data element relationship.

a. Completing the Data Analysis Summary.Note that the Data Analysis Summary shown in Figure 2-5 is divided into three sections. The Input Section (left side) provides space for recording all the inputs within the scope of the study; the Output Section (right side) provides spaces for recording all output documents within the scope of the study and lastly, the Elements of Data Section (center portion) permits recording all data elements in both the inputs and outputs.

(1) The Input Section. To complete the input section of the DAS, the analyst must -

(a) Obtain the titles, document type codes, and frequencies from the input side of the previously completed grid chart and enter these items in the appropriate spaces. For ease of reference when relating documents to data elements, it is best to make these entries working from the center outwards (right to left).

(b) Obtain the peak volume for each input document from the DADS and record it in the appropriate block.

(c) Obtain the general flow reference number for each document from the flow chart of the present system and enter it in the proper space. The general flow reference number is the page number where the document appears on the flow chart of the present system.

(2) The Output Section. To complete the output section of the DAS, the analyst must -

(a) Obtain the titles, document type codes, and frequencies from the end product side of the previously completed grid chart and enter these facts in the proper spaces. Make these entries from left to right.

(b) Obtain the peak volume for eacn output from the DADS and enter it in the appropriate block.

(c) Obtain the general flow reference for each document from the flow chart of the present system, and enter it in the proper block.

DATA ANALYSIS SUMMARY

1 of 3 SHEETS

2. AREA: Stock Records Processing

1a INPUT										3a. OUTPUT									
				Backorder	Due Out	Inventory File	Stock Receipt	Requisition	← DOCUMENT NAME for Input & Output Data →	Inventory File	Due Out	Backorder	Shipping Order	Stock Transfer	Purchase Order				
				2	1	1	0	0	← TYPE DOCUMENT →	1	1	2	3	3	3				
									← GENERAL FLOW REFERENCE NUMBER →										
				250	600	55K	100	2K	← PEAK VOLUME →	55K	600	250	15K	40	120				
				W	W	AR	D	D	← FREQUENCY →	AR	W	W	D	AR	D				

4. ELEMENTS OF DATA

1b									DATA CODE	DESCRIPTION	NO. CHAR	A/N	DEC POS	3b.									
										Depot													
					☒		☒	☒	I	Depot Name	20	AN			☒								
							☒	☒	I	Depot Street Address	20	AN											
							☒	☒	I	Depot City & State	20	AN											
										Requestor													
					☒			☒	R	Name	20	AN			☒		☒						
					☒			☒	R	Address	20	AN			☒		☒						
					☒			☒	R	City & State	20	AN			☒		☒						
								☒	R	Activity Account Number	4	N					☒						
							☒	☒	I	Supply Depot Code	1	A					☒		☒				
				☒	☒			☒	I	Requisition Number	6	N			☒	☒	☒						
								☒	N	Requisition Date	7	AN											
								☒	G	Date Requisition Received	7	AN											
				☒				☒	R	Control Number	6	N				☒							
								☒	R	Line Number	2	N					☒						
				☒	☒	☒	☒	☒	I	Stock Number	11	N		☒	☒	☒	☒	☒	☒				
				☒	☒	☒	☒	☒	I	Nomenclature	20	AN		☒	☒	☒	☒	☒	☒				
								☒	R	Authority	7	AN											
								☒	R	Quantity Requested	3	N					☒						
				☒	☒	☒	☒	☒	I	Unit of Measure	2	A		☒	☒	☒	☒	☒	☒				
						☒		☒	D	Unit Price	7	N	2	☒									
								☒	R	Total Price	7	N	2										
								☒	G	Availability Check	1	AN											

5. DATE 16 DEC 84

6. PREPARED BY R. T. JUMPTHRU

FBH FORM 32-0-17, NOV 83
(ATSD-CM)

Figure 2-5.

(3) The Elements of Data Section. This section of the DAS is completed for each document by using the information contained on the reverse side of the DADS. A complete description of each data element is recorded here. As each data element is recorded, the analyst enters an "X" under each document that uses this element. A data element is only entered in this section one time, no matter how many documents it appears on. Multiple uses of the data element are shown by the use of an "X" in the appropriate column for those documents in which the element is used. All duplication of like items should be eliminated. In some cases, the names given to the same data element may be different, and another interview may be necessary to resolve the problem.

(a) Description. Show the name of the data element here, using, if possible, the standard data element name as defined in AR 18-12.

(b) Number of characters. Record the maximum number of characters shown for the item on the DADS.

(c) A/N. In this column, the analyst shows the types of characters found in the data element (alphabetic, numeric, alphanumeric). This data is transferred from the DADS containing the data element.

(d) Number of Decimal Positions. In the column labeled "No of Dec Positions" the analyst enters the number of decimal positions from the DADS. For example, if a data element calling for the unit price to the exact penny were used, it would require a "2" in this column indicating two positions following the decimal. Any characters counted in this column have been previously counted in the "No of Char" column.

(e) Data Code. In order for a proper analysis of the data elements to be accomplished, a data code must be assigned to each element to describe how that element is used in the system as a whole. Care must be exercised to consider the whole system, and not an isolated occurrence when assigning codes, for this could lead to miscoding and misunderstanding of the data element. Data codes are assigned based on the following definitions:

1. Fixed Elements of Data. These data elements are considered permanent or semipermanent in nature, and they usually change very little. There are two types of fixed data elements.

a. Identifying (I). These elements will either name, or otherwise uniquely identify a document, record, or other entity to the system.

b. Quantitative (Q). These are fairly constant elements which are used in arithmetic operations (calculations), or in measuring.

c. Both "I" and "Q" data elements, because of their relative permanence, lend themselves to inclusion on master files within the system. A high percentage of the data elements found in the master files for any system will normally have been classified as "I" or "Q" by a systems analysis team.

2. Variable Elements of Data. Variable elements of data are continuously changing in nature, and are either the results of some action or calculation, or are used to convey information into the system.

a. Reported (R). This type of data enters the system on code "O" documents. "R" data elements are distinguished by the fact that they always enter the system from an outside source, and by the fact that they are data coming into the system for the first time. Source documents (code O) normally consist of a number of "R" data elements, and some "I" fields which serve to identify the document and link it to the proper master file record.

b. Generated (G). These elements result from arithmetic operations (calculations), or are used to indicate that some action has been taken.

3. Constant (C). The final type of data element is the constant. Constants are normally items in the nature of administrative data items used for repetitive processing, constant items in output reports, or mathematical constants used repeatedly in processing. Constants are stored in the computer, or are entered via the console typewriter or control card.

b. Analyzing the Completed Data Analysis Sheet. A completed Data Analysis Sheet will show the analyst input redundancy, output redundancy, and areas for file consolidation.

(1) Input Redundancy. By examining the "X"s on the input side of the data analysis sheet for each element of data, the analyst can draw certain conclusions. For example, if he finds that two or more documents contain common data elements, he can consider the possibility of combining the documents by modifying one of them. Thus, a reporting requirement could be eliminated. Certain data elements fed into a data processing system need not be reported again until they are changed. Examples of these are quantitative items such as unit price, hourly rate, units of measure, etc. Elements such as these would only be reported to the system when there was a change in their value; otherwise, they would be stored in a master file.

(2) Output Redundancy. When more than one "X" is discovered for a data element in the output section, the analyst should question whether it is required in each document. Certain identifying elements may be required in many documents; however, two or more X's following a specific data items can help the analyst in pinpointing areas where output documents may possibly be combined or eliminated. Here again, the analyst would apply the same principles used in the input area for combining data elements to reduce the number of required documents.

(3) Areas for File Consolidation. Many files are frequently needed within a manual or punched-card system to accomplish the data processing mission. However, through the use of ADPE with its capacity and speed, many elements of data from several files may be combined or grouped in one file. The Data Analysis Sheet provides the analyst with a guide to which of the files in use may be combined.

SECTION 4: FLOWCHARTING

2.37. Introduction & Definition. In the early years of computer programming, a programmer began writing a program by immediately coding the first instruction that came to mind, followed by the second instruction, and so on, until the last instruction had been coded. As the complexity of programs increase, it became apparent that some method was needed to organize the solution to a problem before the coding began. Symbols were used to graphically represent a solution. This early design methodology was called 'Flowcharting'. So important and widespread was its use that the American National Standards Institute (ANSI) published a set of flowcharting standards representing various operations that could be performed manually, or with the use of a computer.

Although the flowchart was a valuable tool to represent logic within a program, it was found that programmers applied their own set of logic and coding rules. Programming was an individual activity. Indeed, some programmers were very protective of their programs, not allowing anyone to see them. The programmer was usually not aware of the overall design of the program until the flowchart was actually completed. Programs were more or less patched together. There was neither an organized structure nor a methodology to approach the program design. Such an attitude was not one which resulted in programs that were reliable and easy to modify and easy to maintain. Soon it became apparent that these traditional techniques had to be changed from an art into a science. An 'ego-less', problem solving approach evolved into what is know today as 'Structured Analysis', 'Structured Design', and 'Structured Programming'. Using this disciplined approach, the final product is no longer an extension of an individual, but rather a product which can be examined, analyzed and critiqued, to make it better. The preparation and use of flowcharts has become an integral part of a 'structured' collection of guidelines for distinguishing between good designs and bad designs during the development life cycle of a data processing system.

The systems development life cycle frequently begins with a user. Having perceived a need or desire for a new system, or dissatisfaction with an existing one, a user carries on a dialogue with a person typically known as a 'Systems Analyst'. The function of the systems analyst is to study the user's current system (if there is one), interview hordes of clerks in user's departments, collect documents and gather information about the types of data elements used to support department functions. The final product of the analysis is a set of functional requirements that describe, in precise terms, without the use of computer jargon:

1) The objectives of the analysis,

2) Definition of the problem,

3) Feasibility study,

4) Cost effectiveness study,

5) User inputs and required outputs,

6) System constraints, and

7) Processing required (calculations and algorithms).

Sometimes a flowchart is included with the other requirements. This flowchart is not likely to explain decisions or sequencing operations, but only those structures, procedures or data flows that an analyst feels others should be aware of. Ideally however, the systems analyst should not design a system. This is somewhat impractical since the systems analyst frequently makes unconscious design decisions all during discussions with a user. It is also unrealistic because the feasibility study and cost effectiveness study require the analyst to make some preliminary decisions about the size, power and cost of computer equipment (or equipment needed to support a manual system), the cost of materials and supplies, personnel, training, and operating overhead. Aside from this, the functional requirements should not specify manual or computer oriented design decisions, like:

1) Number of job steps, number of programs, sorting or merging, etc., that may be needed to implement the solution to the problem,

2) Procedural details, logic or types of controls,

3) Document layouts, record layouts, access techniques, file and storage requirements, or data management software,

4) Number and type of intermediate files to be passed between programs,

5) Type of hardware, and hardware limitations, and

6) Programming language capabilities or limitations.

When completed and approved, the set of functional requirements is turned over to a job function commonly called 'Systems Design'. An individual, called a 'Systems Designer' accomplishes what some organizations call 'General and Detailed Systems Design'. The final product of the systems designer is a set of documents describing the structure of a system and the program specifications (or specifications needed to implement a manual system).

A significant portion of these documents will consist of flowcharts. A flowchart is a design tool; a model of a system. It provides a means of recording, analyzing and communicating problem information. Embedded within the flowchart, in symbolic form, is the explanation of plan or strategy and specific design parameters that will be needed to technically solve a problem. The flowchart can be analyzed, critiqued, and evaluated in terms of making the design better.

There are many types and uses of flowcharts. The format and amount of detail closely parallel the systems development life cycle and the audience to which the flowchart is directed. Initially, the designer will develop an overall solution to the problem. The associated flowchart is called a 'System Chart'. As the design progresses, more and more technical decisions are made. The designer must progressively organize those decisions, making them more explicit and technical, and less implicit and artful. This effort is called 'Detailed systems Design'. The associated flowcharts contain more detail. The designer's orientation can motivate him to develop a transaction-centered flowchart, a procedure-centered flowchart or one of several other modular organizations. The primary user of these documents is a programmer, who has the responsibility to implement the specifications defined by the systems designer.

When traditional design techniques are used, the programmer uses the flowcharts developed by the systems designer to produce other flowcharts in still greater detail. These are called 'Logic Flowcharts' or 'Detailed Flowcharts'. However, new design and documentation formats have been developed in conjunction with structured design techniques. When used, the systems design is divided into major tasks called modules. The intent is to design a system where the modules are small, easily related to the solution to the problem, and relatively independent of one another. Structured design achieves this by focusing attention on the proper partitioning of the solution to the problem, and by the proper arrangement of the modules in the system, early in the design stage. When these new techniques are used, the detailed flowchart is replaced by one or more 'structure charts'. One type of structure chart is called a Hierarchy Chart because of its similarity to an organizational chart. When finished, the structure chart shows all modules required as well as the interconnections between them. The skill needed to prepare a structure chart is usually provided by the system designer. However, it is often done by a programmer who is able to make detailed design decisions. Later on, the modules can be coded by the programmer, or other charts (including the detailed flowchart) can be prepared so that the job can be handed over to a coder. (In the case of the manual system, of course, no computer coding is necessary prior to implementation).

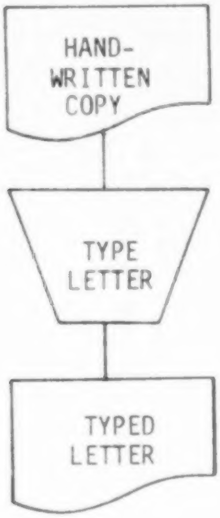
2.38. The systems flowchart. The systems flowchart is primarily the tool of the systems designer. It is a marcro-level model that shows all inputs and outputs to a system, as well as the processing units (such as job steps, computer runs, manual operations, and so on). It can be used to explain or modify an existing system or provide the basis for developing a new system. The systems flowchart does not contain any procedural details concerning the nature of the processing involved. Instead, the systems flowchart concentrates on tracing all inputs and outputs through a manual or automated system, in a time related sequence. The basic format of a systems flowchart is:

SHOW ALL INPUTS

IDENTIFY THE PROCESSING

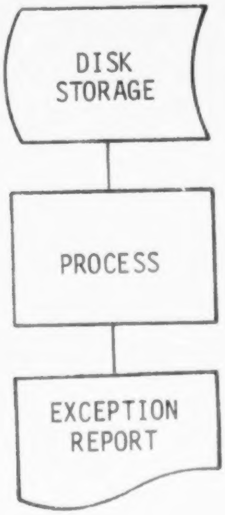
SHOW ALL OUTPUTS

For Example: A systems chart that represents the function of typing a letter from hand-written copy would look like this:



Input to the system is the hand-written copy. The manual process symbol represents the process of typing the letter; the output, or results of processing, is the typed letter. The systems chart makes no attempt to explain the detailed steps or decisions that a typist is likely to make in the process of providing the output from the input. Note the use of the document symbol (an ANSI standard) to represent the input and output, and the manual processing symbol (another ANSI standard). Other ANSI symbols, and their meanings are listed at the end of this section.

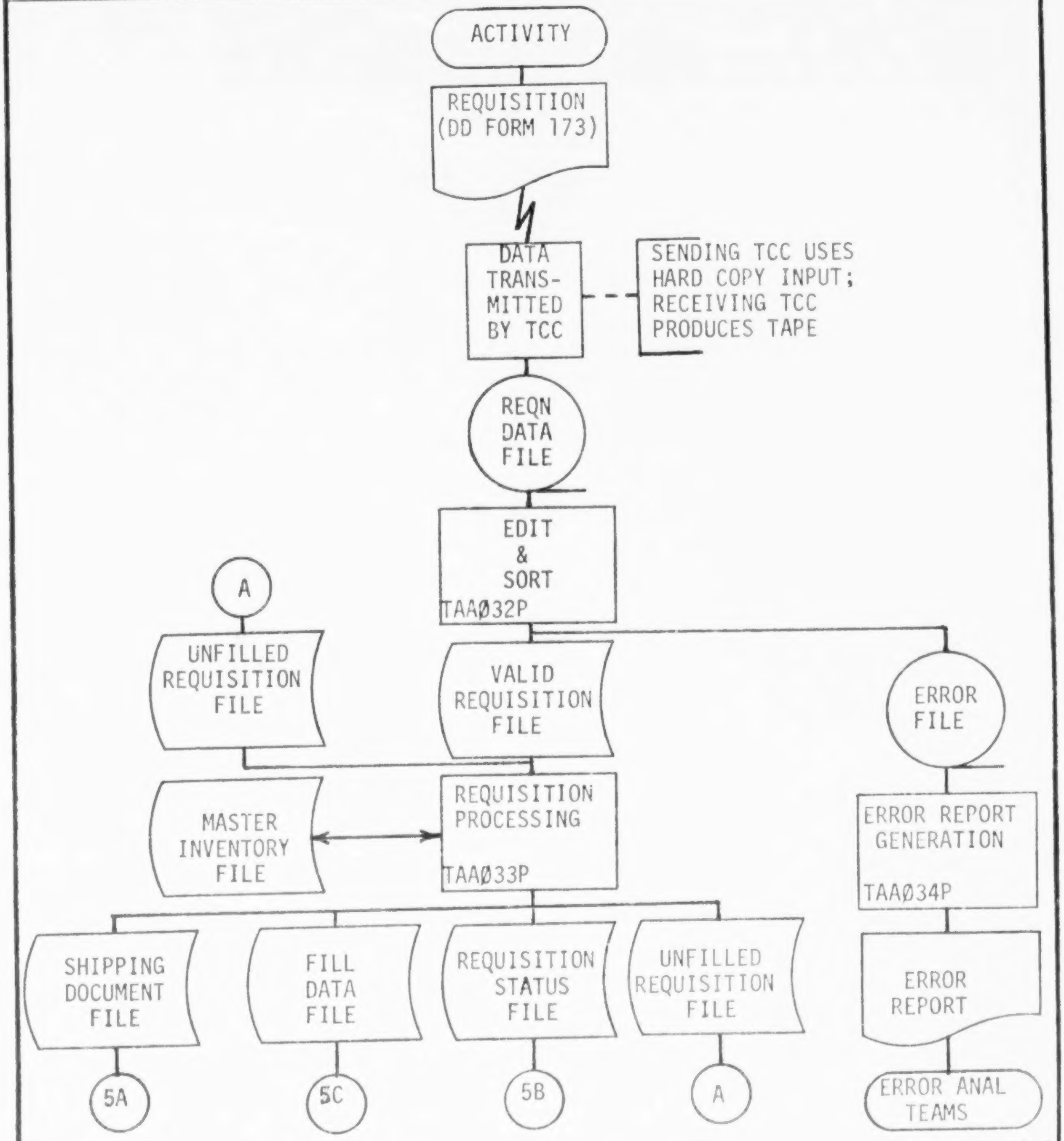
An example of a computer system that extracts certain information from a magnetic disk storage device so that an exception report can be printed would look like this:



Notice the use of the ANSI standard symbol to represent magnetic disk input. Next, since the process is an automated one, a rectangle is used, rather than the manual symbol shown in the previous example. The processing will be a printed exception report.

Most systems charts identify more than one process. The systems chart on the next page shows part of a proposed stock control system. Notice that four separate processes are shown: Data transmitted, edit & sort, requisition processing and error report generation. The complete systems chart required eight pages of documentation. This is just one page of it. Notice the use of on-page and off-page connectors (the small circles containing numbers and letters). Corresponding connectors would appear on the other pages. This flowchart is read following the usual rules for reading printed material: Top to bottom, left to right. The various symbols, and their meanings, can be found at the end of this section. This flowchart was drawn on DA Form 4735 Logic Chart, which is a documentation standard. These charts, prepared by a systems designer, would be reviewed, analyzed, and critiqued so that the final product is a model of the best possible design.

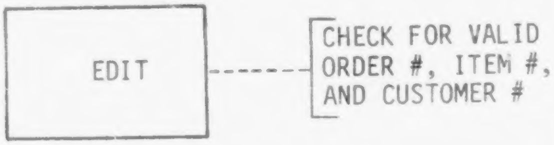
LOGIC CHART For use of this form, see TB 18-111; the proponent agency is CSA.		1. DATE YYMMDD
2. ID TAAD03	3. TITLE PROPOSED STOCK CONTROL SYSTEM (TAA-T99) DAILY REQUISITION BATCH PROCESSING	


ACTIVITY
REQUISITION (DD FORM 173)
DATA TRANSMITTED BY TCC
SENDING TCC USES HARD COPY INPUT; RECEIVING TCC PRODUCES TAPE
REQN DATA FILE
EDIT & SORT
TAA032P
A
UNFILLED REQUISITION FILE
VALID REQUISITION FILE
ERROR FILE
MASTER INVENTORY FILE
REQUISITION PROCESSING
TAA033P
ERROR REPORT GENERATION
TAA034P
SHIPPING DOCUMENT FILE
FILL DATA FILE
REQUISITION STATUS FILE
UNFILLED REQUISITION FILE
ERROR REPORT
5A
5C
5B
A
ERROR ANAL TEAMS

4 of 8

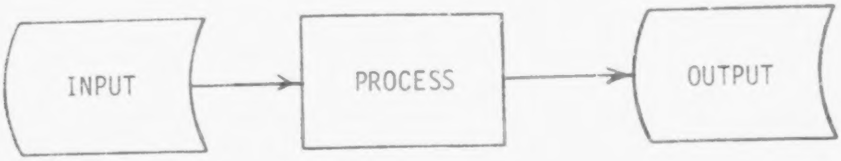
SYSTEMS FLOWCHART CONSTRUCTION INVOLVES FOUR STEPS:

1) Identify all major processing tasks within the system. Processing can be accomplished using a computer, or manually (at human speed) with or without the support of equipment such as: Typewriters, Word Processing Systems, Data Entry Devices, and so on. For Example: The following example suggests a computer editing task. The use of some key word inside the symbol provides the reader with a clue to the major task involved. Further explanation can be optionally provided, using an information symbol.



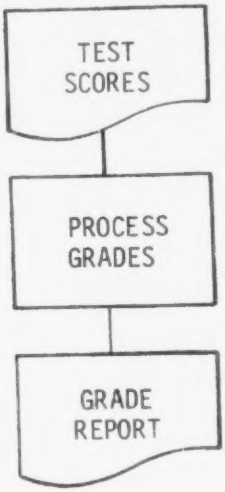
2) Identify all the inputs and outputs within the system. An input is that which is required by a major processing task. An output is the result of processing. Outputs exist as a final product, or they become inputs to support other processing tasks.

3) Identify the sequence of actions. The sequence is defined by connecting the inputs, processes and outputs together using flow lines. At times, the sequence of events may be obvious. For more complex designs, there may be more than one sequence of events that could provide the same final product. The skill of the systems designer is important in determining the best overall design. Inputs must flow into a processing unit. Outputs must flow from a processing unit, possibly into another process. Inputs never flow directly into other inputs; outputs never flow into other outputs. Processing never flows directly into another processing function. Each processing function separates inputs from outputs.

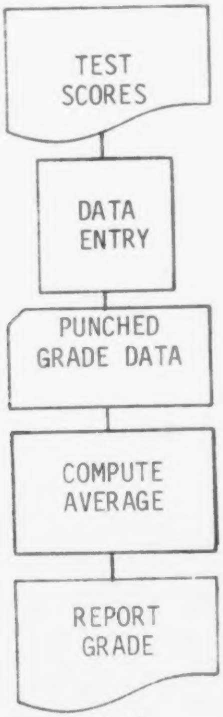


4) Use standard ANSI symbols. The use of standard symbols reduces the need to explain the logic and strategy embedded in a flowchart. Also, there will be instances where other systems analysts and users will become involved in, or assume responsibility for part or all of a system. Others should be able to arrive at the same conclusions and form the same understandings concerning the flow of inputs and ouputs, related processing, and the sequence of events. Standard symbols help eliminate misinterpretation.

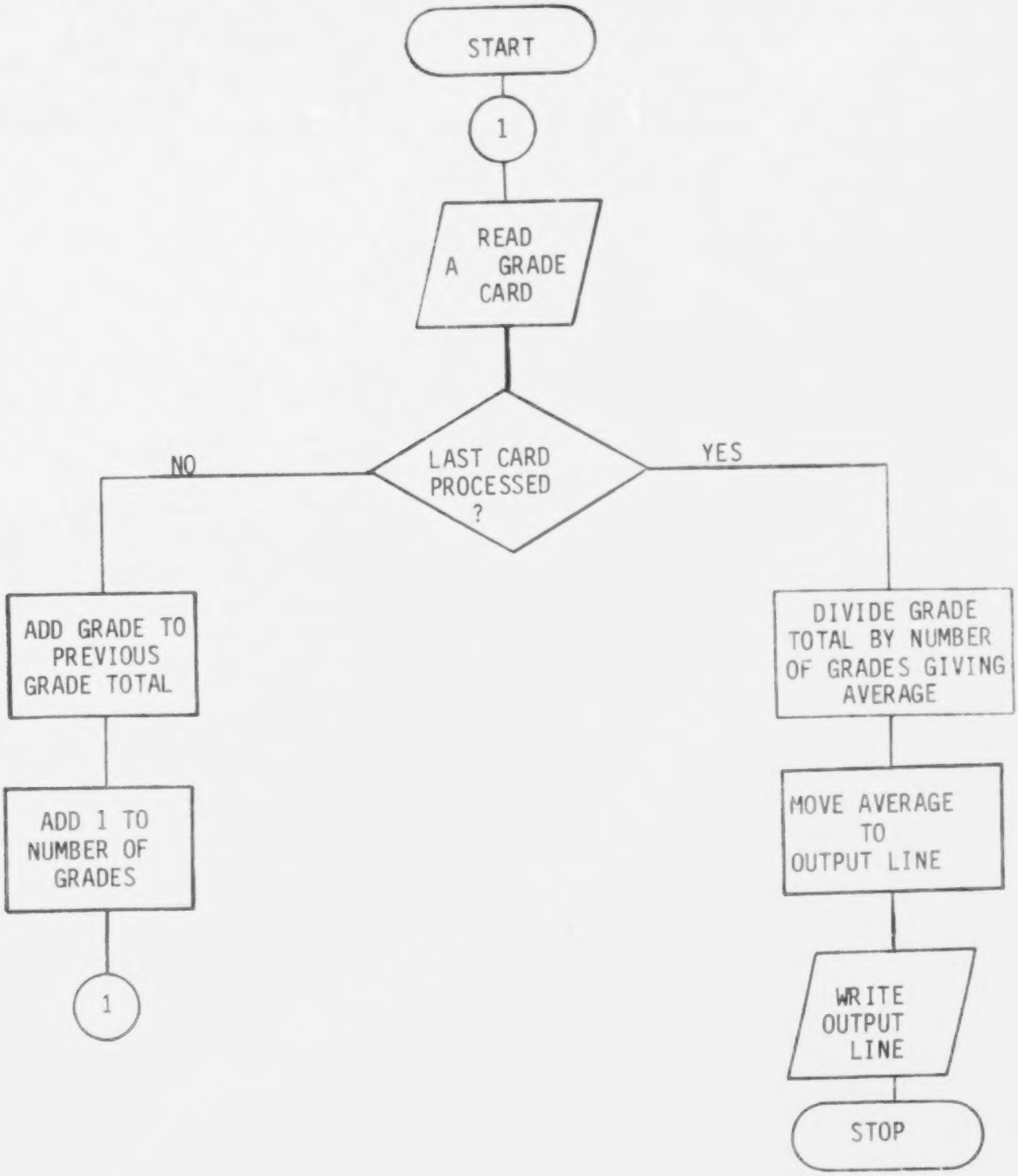
2.39. A detailed flowchart is a procedure-centered flowchart (sometimes called a Logic Chart), that graphically represents detailed procedures of manual or automated functions. It evolves out of a systems flowchart. To illustrate the construction of a detailed flowchart, consider the following example. Major Hal I. Tosis wishes to determine the average grade for each student based on 35 tests given in a Data Processing course. Once the average grade is calculated, a student grade report is to be printed. The systems chart could be drawn as:


TEST SCORES
PROCESS GRADES
GRADE REPORT

Or a system chart could show two processing steps, instead of one:

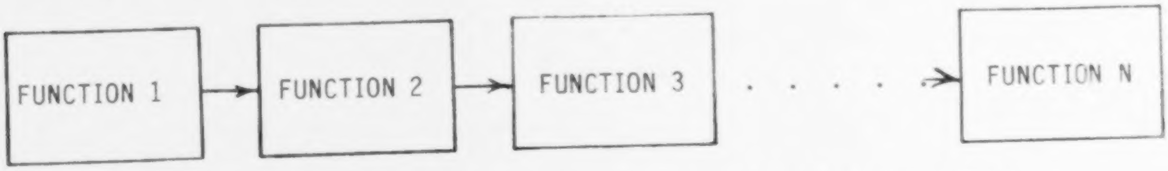

TEST SCORES
DATA ENTRY
PUNCHED GRADE DATA
COMPUTE AVERAGE
REPORT GRADE

A detailed flowchart (Logic Chart) showing the automated steps represented by the process called 'Compute Average', might look like:

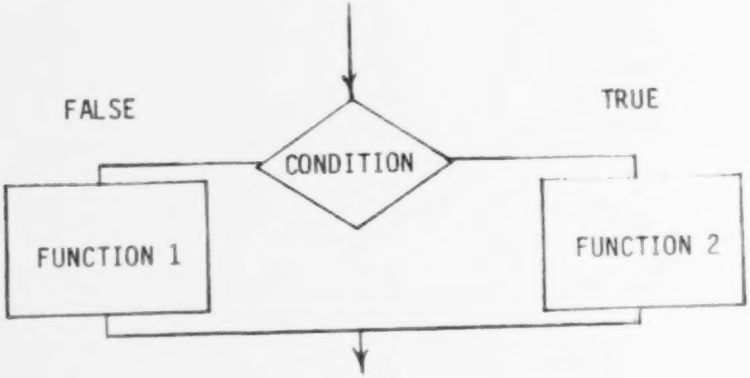

START
1
READ A GRADE CARD
LAST CARD PROCESSED ?
NO
YES
ADD GRADE TO PREVIOUS GRADE TOTAL
ADD 1 TO NUMBER OF GRADES
1
DIVIDE GRADE TOTAL BY NUMBER OF GRADES GIVING AVERAGE
MOVE AVERAGE TO OUTPUT LINE
WRITE OUTPUT LINE
STOP

Each rectangle represents a specific computer process. The triangle represents a decision that will determine which of the two alternate paths is to be followed. Note the use of a connector (with the number one inside a circle) to indicate a repeating process.

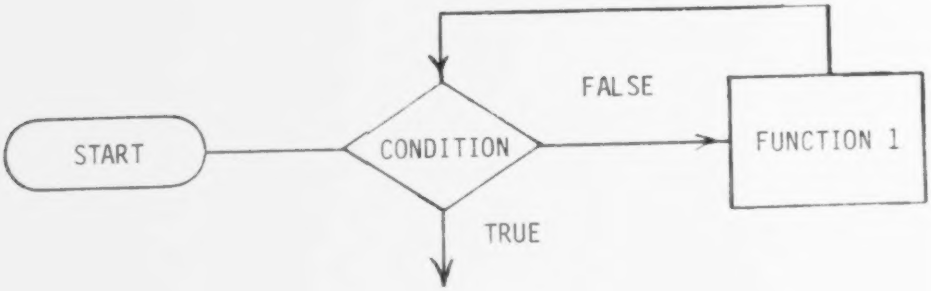
The theory of structured programming is: No matter how complex logic becomes, any program can be written using three basic control structures: 1) Sequence, 2) Selection and 3) Iteration. A detailed flowchart can be used to illustrate these structures. The sequence executes one instruction right after another, in the sequence listed. In the next illustration, each rectangle represents an instruction. For an automated system, an instruction could be to add two number, or move data from one storage location to another location.


FUNCTION 1
FUNCTION 2
FUNCTION 3
FUNCTION N

Another structure is called a 'Selection Structure' or "If-Then-Else' Structure. This structure is used for conditional statements. A condition is tested to determine whether it is true or false. If the condition is true, one path is followed. If the condition is false, an alternate path is followed, shown in the next example.


FALSE
TRUE
CONDITION
FUNCTION 1
FUNCTION 2

The third structure, iteration, is used to illustrate looping. Looping means that one or more instructions are to occur repeatedly, so long as a given condition remains true (or false). When the condition changes, the looping process is terminated.


FALSE
START
CONDITION
FUNCTION 1
TRUE

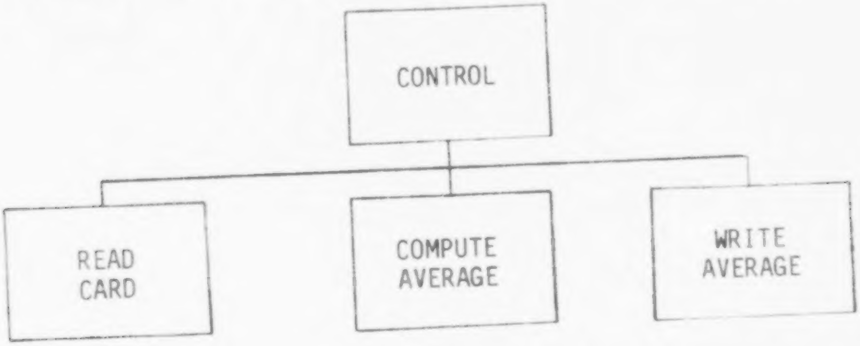
When used separately, or in various combinations, these three structures can be used to solve any programming logic problem. The detailed flowchart can be a useful tool to illustrate the solution to a problem because flowcharts help:

1) State the logic in clear terms;
2) Show essential details;
3) Remove fuzzy thinking;
4) Clarify transfers of control;
5) Eliminate accidential omission of detailed steps.

In spite of the advantages of detailed flowcharts, they do have some limitations. In the design stage, flowcharts can be difficult to plan and draw especially when the logic is complex. It is easy to leave out important details. Once created, detailed flowcharts can be difficult to follow and difficult to change. Also, certain detailed flowcharts cannot be coded into a computer, as written. Consequently, they can be a poor guide for coding, and usually incomplete documentation. A decision table is often the better tool. But most damning, detailed flowcharts do not help to simplify large or complex designs. They do a poor job of solving the dilemma of how to extract functional specifications from the user and can mislead one into thinking that structured analysis and design techniques are equivalent to simply providing detailed flowcharts.

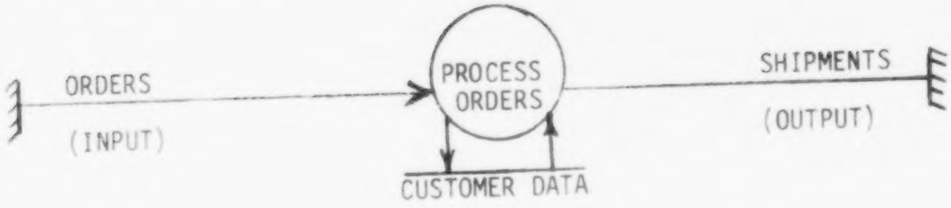
2.40. Structure Charts. When using structured programming techniques, it is really not always necessary to show every detailed step before the coding begins. The three basic control structures that show 'How' certain functions are to be performed can often be applied directly to the coding process by a programmer. But there is no assurance that the final coded product will be a 'Structured Program' that is easy to code, reliable, and easy to maintain and modify. Indeed, if the program has any complexity to it at all, attempts to achieve these goals will be, in all probability, a failure. What is needed is some method of showing 'what' functions are to be performed. A 'Structure Chart' is a design tool that assists a systems designer and programmer in organizing the solution to a problem. One type of structure chart is called a 'Hierarchy Chart'. They are popular because they are similar to organization charts, and therefore, easy to relate to. When completed, a Hierarchy Chart provides, in symbolic form, a picture of all the pieces of the solution to a problem. The designer can view the shape of the entire structure - width and depth, in order to judge the goodness of the design. Highly complex pieces of the design might need to be flowcharted in detailed. But this will be done with the awareness of 'how' such detail fits into the overall design. More than likely, less complex pieces can be coded or implemented directly without the need for additional detailed flowcharts.

An example of a structured Hierarchy Chart representing the solution to the problem of providing student grade averages would look like this:



The depth of the Hierarchy Chart gives the designer clues to the size and complexity of the system. If the Hierarchy Chart has three or four levels, the program is likely to be a simple one - perhaps containing no more than 100 computer instructions. A depth of 10 to 12 would suggest a program of moderate size. A massive program would have 50 levels within a hierarchy. A 100 statement program representing a Hierarchy Chart contains a dozen levels would indicate excessive zeal on the part of the designer. A 5,000 statement program with a depth of two or 3 would suggest very poor separation of functions. Similar conclusions could be made observing the width of the Hierarchy Chart. In general, the Hierarchy Chart should be about as wide as it is deep.

2.41 Data Flow Graph. A systems designer most often studies the flow of data through a system or program. A method is needed to restate the functional requirements or system specifications that will emphasize the data flow and ignore the procedural aspects of the problem. The transaction-oriented technique is called a data flow graph. (Also called a bubble chart). A data flow graph defines the functions that are to be accomplished, the data elements needed to accomplish those functions and the relationship between data elements and function. It is similar to a systems flow chart, and often a substitute, except that the data flow graph shows the logical flow of data, whereas the systems chart shows the physical flow of data. Like the systems chart, the data flow chart does not show the flow of control or any procedural details. The elements of a data flow graph are called 'transforms', represented graphically by a circle (or bubble). The circles represent the transformation of data from one form to another. At the detail design stage, each transformation will be defined by a module, a program, or an entire system. Each transform 'bubble' is identified using a verb plus a specific object which describes the function to be performed. The data elements are represented by labeled arrows connecting one transform to another. The following example shows the basic elements of a data flow graph using one input and one output stream.


ORDERS
(INPUT)
PROCESS ORDERS
SHIPMENTS
(OUTPUT)
CUSTOMER DATA

The guidelines for preparing a data flow graph are:

1) Provide sufficient detail to model the system. If in doubt, show more detail rather than less.

2) Show all inputs and outputs in the system.

3) Show the existence of error paths that separate good data from bad data; avoid showing simple error paths to the outside world.

4) Do not show control logic, initialization, termination, procedures, decisions, loops or other entanglements.

5) Avoid computer jargon.

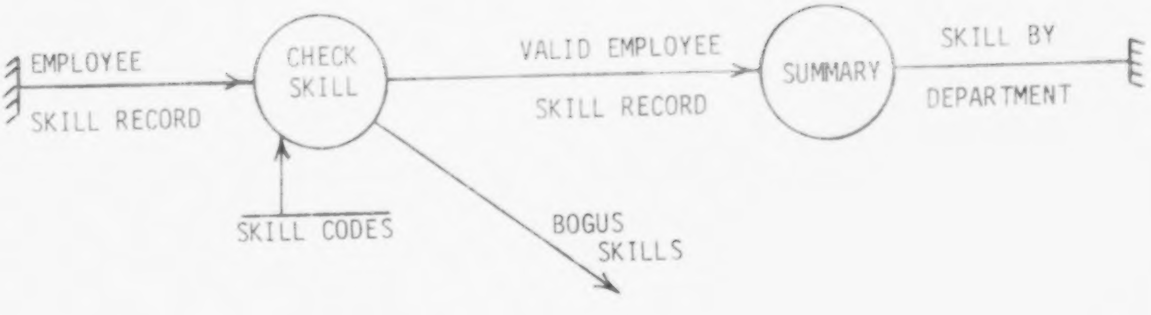
6) Name each transform bubble using a meaningful verb followed by an object.

7) Introduce data at the point it is needed, not before.

8) Do not think in terms of files, records, data bases, hardware or programming languages.

9) Begin with the input, working the way downstream through successive transforms to the physical output; working upstream from outputs toward inputs works just as well.

Another data flow graph, containing two transforms, looks like this:

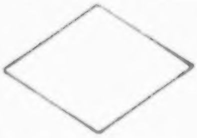
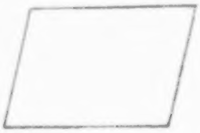
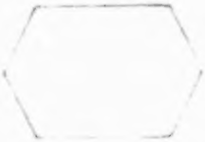
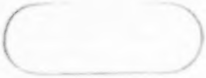
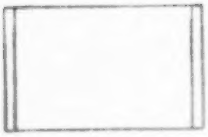
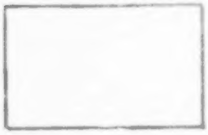
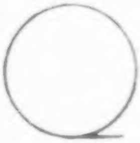
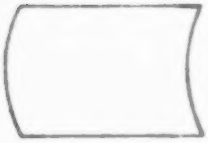
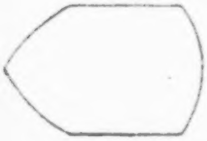

EMPLOYEE
SKILL RECORD
CHECK SKILL
SKILL CODES
VALID EMPLOYEE
SKILL RECORD
BOGUS SKILLS
SUMMARY
SKILL BY
DEPARTMENT

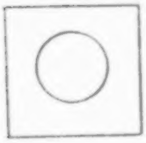
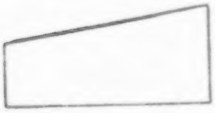
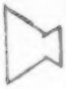
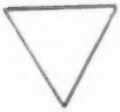
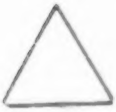
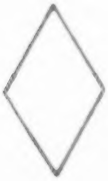
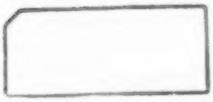
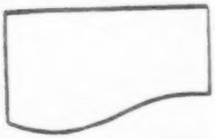
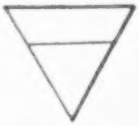
The arrow pointing to a base line identified as 'skill codes' shows that data is being extracted from a database of skill code information. Notice the direction of the arrow. An arrow pointing downward would be used to show that the skill code information is being 'updated' or changed. Arrows in both directions would suggest both update and retrieval type flows of data. The arrow representing the flow of 'Bogus Skills' data represents the existence of an error path that separates good data from bad data. Its use is optional.

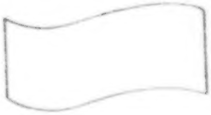
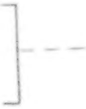
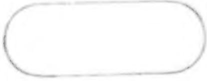
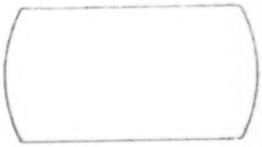
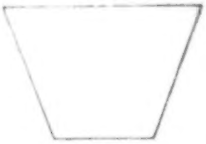
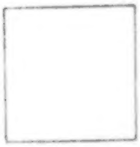
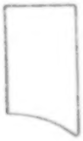
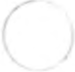
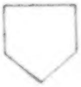
All flowcharting requires other supporting documentation. In general, this supporting documentation can be thought of as being a 'data dictionary'. A data dictionary is document or collection of documents containing information about data. In the case of a data flow graph, a data dicitonary would be used to support each primitive (transform). This dictionary would contain information on the data-names described in the data flow graph, the files that are used, and the data elements. In addition, a set of functional specifications for that particular transform, would also be included.

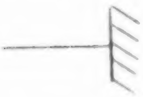
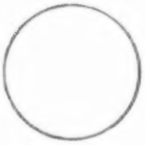
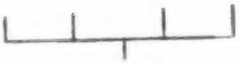
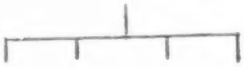
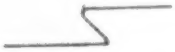
One of the purposes of a data dicitionary is to force out ambiguity and inconsistency in the data. Standard data-names, and data formats are established. Of course, when it is necessary to consider the procedural aspects of the design, documents such as file specifications (DA Form 4737) and record specifications (DA Form 4738) will become important, and are used regardless of the type and extent of the flowcharts.

2.42 ANSI FLOWCHARTING SYMBOLS

		USES		
SYMBOL	MEANING	SYSTEM FLOWCHART AUTOMATED/MANUAL	DATA FLOW DIAGRAM	LOGIC CHART
	DECISION			X
	INPUT OUR OUTPUT			X
	AUTOMATED PREPARATION			X
	START/STOP OR INTERRUPT			X
	PREDEFINED PROCESS			X
	COMPUTER PROCESS	X		X
	MAGNETIC TAPE	X		
	INPUT OR OUTPUT ON MAGNETIC DISK	X		
	CRT, PLOTTER, OR OTHER VISUAL DEVICE	X		

SYMBOL	MEANING	USES		
		SYSTEM FLOWCHART AUTOMATED/MANUAL	DATA FLOW DIAGRAM	LOGIC CHART
	FLOPPY DISK (NOT ANSI STANDARD)	X		
	ON-LINE KEYBOARD	X		
	COM (NOT ANSI STANDARD)	X		
	MERGE	X		
	EXTRACT	X		
	SOFT - ARRANGE IN A SEQUENCE	X		
	PUNCHED CARD	X	X	
	DOCUMENT	X	X	
	OFF-LINE STORAGE	X	X	

SYMBOL	MEANING	USES			
		SYSTEM FLOWCHART AUTOMATED/MANUAL		DATA FLOW DIAGRAM	LOGIC CHART
	PUNCHED TAPE	X	X		
	INFORMATION OR COMMENTS	X	X		X
	DISPOSITION OF OUTPUT OR SOURCE OF INPUT		X		
	KEYING (NOT ANSI)		X		
	MANUAL OPERATION (AT HUMAN SPEED)		X		
	AUXILIARY OPERATION		X		
	TRANSMITTAL TAPE (NOT ANSI STANDARD)		X		
	PAGE CONNECTOR	X	X	X	X
	OFF-PAGE CONNECTOR (NOT ANSI STANDARD)	X	X	X	X

SYMBOL	MEANING	USES SYSTEM FLOWCHART AUTOMATED/MANUAL		DATA FLOW DIAGRAM	LOGIC CHART
	SOURCE/SINK (RAKE)			X	
	TRANSFORM			X	
	DATA FLOW (INPUT)	X	X		X
	DATA FLOW (OUTPUT)	X	X		X
	TELECOMMUNICATIONS LINE	X	X		X

Section 5. DECISION TABLES

2.43. INTRODUCTION. One of the greatest problems facing the analyst is the communication of information. How do you tell someone else what you are thinking? This is a task, especially in the area of logic. One method of communication could be the use of a flowchart or logic chart (logical block diagram). If the analyst attempts to use a flowchart, there may not be enough detail or logic for the programmer. On the other hand, if a logic chart is prepared, the analyst is dictating the actual program construction to the programmer. Neither of these situations is desirable.

This void is filled by the decision table (sometimes referred to as the decision logic table). A decision table is simply a tool for recording decisionmaking processes. Its use is not limited to analyst-programmer communication. It is an effective communications tool and also an important part of the overall systems documentation package.

The decision table is used to show an if-then type of relationship. Or, it can be described in terms of conditions, actions, and the resulting decision rule. In this section, three types of tables will be discussed. They are (1) limited entry; (2) extended entry; and (3) mixed entry. Before these types are discussed, the various parts of a decision table should be examined.

2.44. PARTS OF A DECISION TABLE. As mentioned previously, a table is made up of conditions, actions, and rules. The part that lists the narrative portion of the conditions and actions (the left side) is known as the stub. The part that contains the rules (the right side) is the entry. The top contains conditions and the bottom contains actions. The sections of a decision table are condition stub, action stub, condition entry, and action entry, as shown in figure 2-13. As indicated, the entry portion of the table is subdivided into columns called rules. The illustration in figure 2-13 is called the vertical format and is the most popular form. Examples of alternative forms such as the horizontal format are found in DA Pam 310-11. SIDPERS Users Manuals, Finance Pay and Allowance Tables, and EPMS Task Analysis Papers are all practical applications of decision tables within the Army.

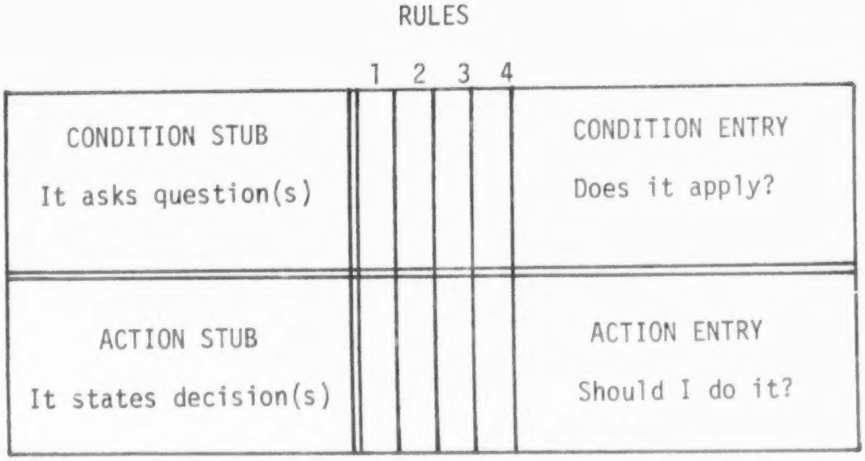

RULES
1
2
3
4
CONDITION STUB
It asks question(s)
CONDITION ENTRY
Does it apply?
ACTION STUB
It states decision(s)
ACTION ENTRY
Should I do it?

Figure 2-13. DECISION TABLE FORMAT

A condition is an event or fact that influences the actions to be taken. An action is any operation or process to be performed. A rule is a single combination of conditions and actions.

2.45. LIMITED ENTRY. This table type requires the entire condition or action to be written in the stub. Rule entries consist of "Y", "N", "X", and "-" or blanks.

a. Y means that the condition is satisfied.

b. N means the condition is not satisfied.

c. Blank or "-" means that the action is taken for either a yes or a no value of that condition, and as such, has no bearing on the action taken.

d. X means to take the action that is listed to the left of the action entry.

e. Y or N with a "-" though it means that the stated rule is an irrelevant or impossible condition based upon the problem.

2.46. EXAMPLE. Let us consider a simple example, and the steps required to perform this task.

SITUATION:

Whether to stay in or get out of the Army depends upon a combination of conditions.

If we get at least an 8-percent pay raise I will definitely stay in the Army. If the pay raise is less than 8-percent but I am on the next promotion list and can get either a reassignment or another MOS producing school, I will stay in: otherwise, I will get out of the Army.

CONDITIONS:

I get at least an 8-percent raise.
I get promoted.
I get reassigned.
I get new MOS training.

ACTIONS:

I stay in.
I get out.

Notice that the conditions have only two values or responses: yes or no. This problem could be represented by a flowchart as shown on the next page (figure 2-14).

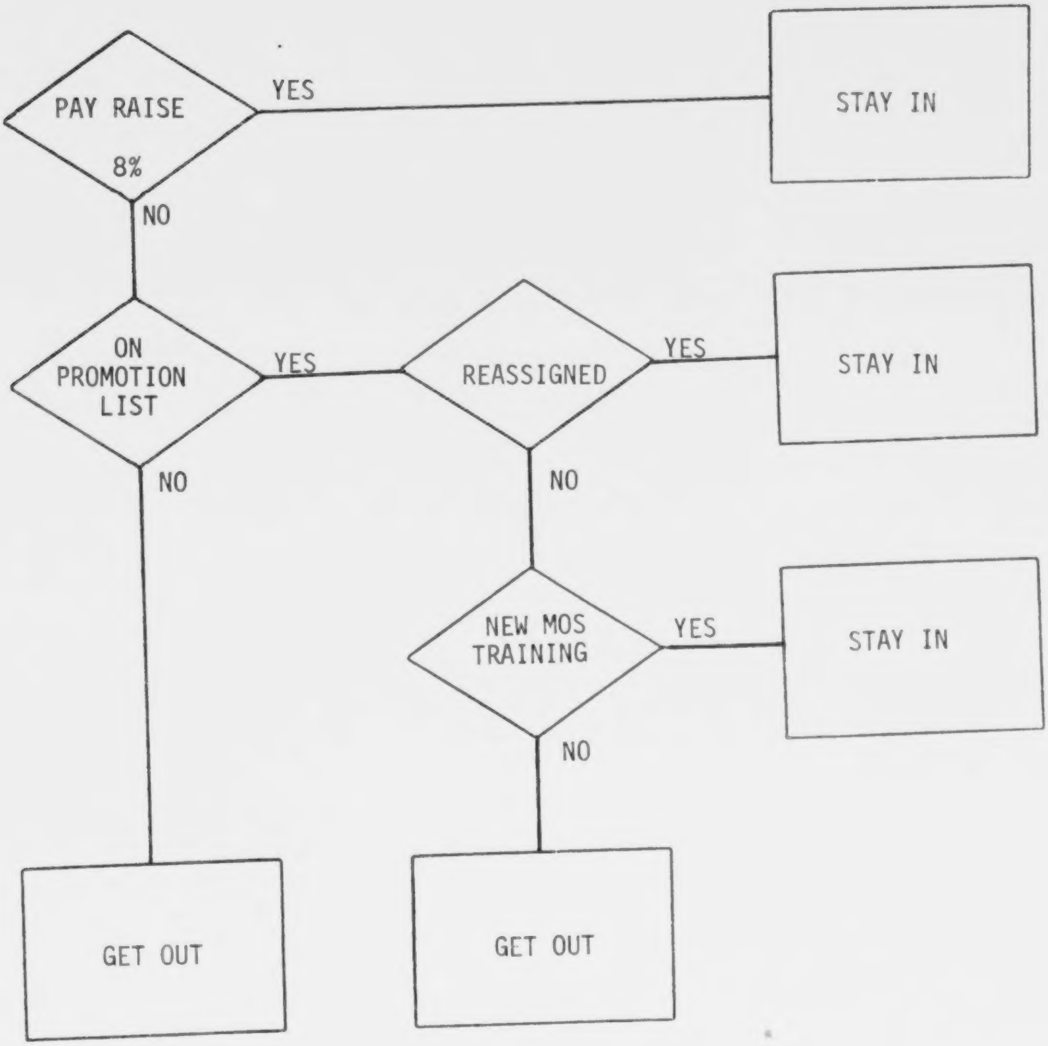

PAY RAISE
8%
YES
STAY IN
NO
ON PROMOTION LIST
YES
REASSIGNED
YES
STAY IN
NO
NO
NEW MOS TRAINING
YES
STAY IN
NO
GET OUT
GET OUT

Figure 2-14.

Five paths may be found in this flow chart. Each path is a decision rule. The corresponding decision table is shown in figure 2-15.

2.47. STEPS IN CONSTRUCTING A DECISION TABLE. After examining the situation and flowchart, the next question might be "how do I construct a decision table?" There are seven basic steps required to perform this task.

best copy available

a. Define and List All Conditions. The problem that is being studied should be examined and all relevant conditions defined and listed. It should be noted that the sequence of conditions does not affect the accuracy of the table, but the sequence does affect its construction and use. A clear and concise condition statement is the best solution. Reference (A), Figure 2-15.

b. Define and List All Actions. Defining and listing all actions provides a check for completeness and problem understanding. The sequence of actions is important. They should be listed in the order in which they are to be performed. Reference (B), Figure 2-15.

c. Complete Condition Entry - Full Matrix. In completing the condition entry, it is necessary to state all of the rules which represent all of the combinations, with no repetition or omission of any combination. This is not as difficult as it might seem. The first step is to calculate the total number of rules required for the conditions. For example; there are four conditions, C1, C2, C3, and C4, and each has two responses. C1 (two values) times C2 (2 values) times C3 (2 values) times C4 (2 values) = 16 rules. For a limited entry table, this can be expressed as 2^n, where n is the number of conditions and 2 is the number of possible answers for each condition.

(1) Knowing the number of rules, the next step is to calculate a group factor. It is obtained by dividing the total number of rules by the values of the first condition. Then take the remainder and divide it by the number of rule values for the second condition, and so on. For example, with 16 rules:

Number of values of C1; group factor 16/2 = 8

Number of values of C2; group factor 8/2 = 4

Number of values of C3; group factor 4/2 = 2

Number of values of C4; group factor 2/2 = 1

(2) The group factors are used to arrange the values of the matrix (condition entry). Each row is produced by repeating each entry the number of times indicated by the group factor. From the preceding example, the first condition would have eight Y and eight N:

Row 1 YYYYYYYY NNNNNNNN

The second row would have four Y, four N, four Y, and four N.

Row 1 YYYYYYYY NNNNNNNN

Row 2 YYYY NNNN YYYY NNNN

LIMITED ENTRY RULES

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
	CONDITION STUB																			(C) CONDITION ENTRY COMPLETE CONDITION ENTRY - FULL MATRIX -
C1	PAY RAISE ≥ 8% (A)	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N			
C2	PROMOTION LIST	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N	N	N	N			
C3	REASSIGNMENT	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N			
C4	MOS TRAINING	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N			
	ACTION STUB																			ACTION ENTRY
A1	STAY IN (B)																			
A2	GET OUT																			
	CONDITION STUB																			(D) ELIMINATE (LINE OUT) ALL CONDITION ENTRIES WHICH DON'T APPLY TO THE SITUATION, OR HAVE NO BEARING ON THE SITUATION
C1	PAY RAISE ≥ 8%	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N			
C2	PROMOTION LIST	~~Y~~	~~Y~~	~~Y~~	~~Y~~	~~N~~	~~N~~	~~N~~	~~N~~	Y	Y	Y	Y	N	N	N	N			
C3	REASSIGNMENT	~~Y~~	~~Y~~	~~N~~	~~N~~	~~Y~~	~~Y~~	~~N~~	~~N~~	Y	Y	N	N	~~Y~~	~~Y~~	~~N~~	~~N~~			
C4	MOS TRAINING	~~Y~~	~~N~~	~~Y~~	~~N~~	~~Y~~	~~N~~	~~Y~~	~~N~~	~~Y~~	~~N~~	Y	N	~~Y~~	~~N~~	~~Y~~	~~N~~			
	ACTION STUB																			(E) COMPLETE ACTION ENTRY
A1	STAY IN	X	X	X	X	X	X	X	X	X	X	X								
A2	GET OUT												X	X	X	X	X			
	CONDITION STUB	COMBINE 1-8								9-10				13-16						(F) CONSOLIDATE -COMBINE RULES HAVING SAME CONDITION ENTRIES AND SAME ACTION ENTRIES (FORM COMPLEX RULES)
C1	PAY RAISE ≥ 8%	Y								N		N	N	N						
C2	PROMOTION LIST	-								Y		Y	Y	N						
C3	REASSIGNMENT	-								Y		N	N	-						
C4	MOS TRAINING	-								-		Y	N	-						
	ACTION STUB																			
A1	STAY IN	X								X		X								
A2	GET OUT												X	X						

(G) CHECK THE TABLE

Figure 2-15.

The third and fourth rows are completed in the same way. They would appear as follows: Reference (C), Figure 2-15.

Row 1	C1	8	YYYYYYYYNNNNNNNN
Row 2	C2	4	YYYYNNNNYYYYNNNN
Row 3	C3	2	YYNNYYNNYYNNYYNN
Row 4	C4	1	YNYNYNYNYNYNYNYN

d. Eliminate Redundant Condition Entries. Utilizing the group factor method to construct a full set of rules for a given situation will usually generate condition entries which do not apply to the situation. Since this is the case, each condition entry within a rule must be evaluated to see if it still applies to the situation. All condition entries (Y's or N's) not applying to the situation must be eliminated (lined through) from the rule. Reference (D), Figure 2-15.

e. Complete the Action Entry. Based upon the valid condition entries for each rule, fill in the action entries which apply to the situation. Analyzing each rule separately, place an "X" in the action entry block(s) of the rule which applies to the action(s) to be taken. Reference (E), Figure 2-15.

f. Consolidate - Form Complex Rules. The formation of complex rules from simple rules can almost be a mechanical process. The two questions to ask are (1) Are the action entries identical? and (2) Are all the condition entry values the same except for one pair? If the answer to both questions is yes, the rules may be combined. Using Figure 2-15, Rules 9 and 10, you can see how this is accomplished.

	CONDITION ENTRIES	COMBINED
Condition 1	NN	N
Condition 2	YY	Y
Condition 3	YY	Y
Condition 4	YN	-
Action 1	XX	X
Action 2		

In the above example, the same action (Action 1) is taken regardless of the response to condition 4 (either yes or no). The "-" is often used to indicate that the condition entry is either yes or no. As shown above, certain steps must be performed to form complex rules from simple rules. They are:

(1) Starting from rule 1, take two rules in sequence, that is (1 & 2), (3 & 4), etc., and compare their condition and action entries. If the entries are as stated in the preceding paragraph, combine them as described in that paragraph and in the example. Continue this comparison and combining for each successive pair of rules; i.e., 1&2, 3&4, 5&6, etc., until the entire table has been analyzed and combined.

(2) Write down the combined rules on your decision table. Reanalyze and combine any of these combined rules following the instructions in step (1). Continue this procedure until all possible rules have been reduced to their basic, unique rule. Reference (F), Figure 2-15.

g. Check The Table. The table should be checked for accuracy and completeness at all steps of development.

If the mathematical group factor method was used to construct the table, it should be checked to make sure that no transportations of condition entries or other errors were made in the condition/action stub construction.

If the logical approach (by guess or "by golly" method) was used, it should be checked to see if all condition entries were accounted for. The logic structure of this method can be checked to a certain extent.

Prior to forming complex rules, count the number of simple rules contained in the table. If the number of simple rules corresponds to the calculated value (2n), the arithjmetic check is complete. This dsoes not prove that the table is correct but it does indic ate that it is probably correct. If the table fails the test, it definitely contains an error.

Remember: All rules, conditions, and actions should be verified at each step of development.

2.48. EXTENDED ENTRY. The limited entry table works very well as long as all conditions take on only two values; however, this is not always the case. To handle these situations, let us examine the extended entry table. In this type of table, the condition stub serves only to identify the variables to be tested, and the condition entries must define the values or states of the variables. Likewise, in such a table, the action stub only names an action, and the action entries give the specifications for the action names. Extended entry format tends to decrease the number of items listed in both the condition and action stubs, and it offers the possibility of considering more than two responses to a given condition. However, it is often necessary to convert extended entry tables to limited entry form if they are to be programmed. Figure 2-16 is an example of the extended entry decision table.

49. MIXED ENTRY. The mixed entry form of table is merely the combination the limited entry form and the extended entry form into a single table. ven though these two forms may be combined, one form must be used exclusively thin each horizontal row of the table. Figure 2-16 contains a sample of the ixed entry form of decision table.

50. SUMMARY. Decision tables are often a vital part of the analyst's cumentation package. They may be especially helpful in spelling out a oblem clearly, systematically, and precisely.

	EXTENDED ENTRY	RULE 1	RULE 2	RULE 3	RULE 4	
	CONDITION STUB					CONDITION ENTRY
C1	IF THE PAY RAISE IS	≥ 8%	< 8%	< 8%	< 8%	
C2	BUT MY CURRENT STATUS IS	------	PROMOTE	PROMOTE	NON-PRO	
C3	AND I WILL BE	------	REASGN	------	------	
C4	OR GET ADDITIONAL	------	------	TRAINING	------	
						ACTION ENTRY
	ACTION STUB					
A1	MY ARMY STATUS WILL BE	STAY IN	STAY IN	STAY IN	GET OUT	
	CONDITION STUB					CONDITION ENTRY RULES 2 AND 3 COULD BE COMBINED TO MAKE THE TABLE MORE COMPACT.
C1	IF THE PAY RAISE IS	≥ 8%	< 8%		< 8%	
C2	BUT MY CURRENT STATUS IS	------	PROMOTE		NON-PRO	
C3	AND I WILL BE	------	REASGN		------	
C4	OR GET ADDITIONAL	------	TRAINING		------	
	ACTION STUB					ACTION ENTRY
A1	MY ARMY STATUS WILL BE	STAY IN	STAY IN		GET OUT	
* *	* * MIXED ENTRY * * * *	* * * *	* * * *	* * * *	* * * *	* * * * * * * * * * * * *
	CONDITION STUB					CONDITION ENTRY
C1	PAY RAISE IS	≥ 8%	< 8%	< 8%	< 8%	
C2	PROMOTION LIST STATUS	------	PROMOTE	PROMOTE	NON-PRO	
C3	WILL I BE REASSIGNED	------	Y	------	------	
C4	WILL I GET MOS TRAINING	------	------	Y	------	
	ACTION STUB					ACTION ENTRY
A1	MY ARMY STATUS WILL BE	STAY IN	STAY IN	STAY IN	GET OUT	

Figure 2-16.

CHAPTER 3 --

Section 1. CONCEP

3.1. INTRODUCTION. The first phase
concentrates on the identification of t
in the user's request for ADP support a
of the procedures used in the current s
the preliminary survey. The conduct of
study team to interview key management
organizational documentation. The seco
detailed analysis, is a full-scale inve
system. This is the fact-finding analy
it is at this time that nearly all of t
techniques are used. After the facts h
recorded, the existing system is analyz

3.2. USER AND SYSTEM REQUIREMENTS.
should be developed which will solve th
At this time, the study team should be
questions. What information is needed?
form? where does it originate? When?
the study teams should be fully aware o
information. At the same time, the tea
requirements the user has for the propo
include all of management's dsesires or
specific information outputs. These sy
are not limited to, the following consi

a. Flexibility and performance.

b. Cost and installation schedule

c. Reliability and maintainabilit

d. Expected growth and life expec

3.3. MEETING REQUIREMENTS. After h
system requirements, the study team mus
most effective use of the available res
material, etc.). The study team should
available as a result of the detailed a
the study team will seek to make the be
which resources are put may not be opti
an organization require the use of the
on the use of resources are known as co
affect the future design efforts should
analysis.

STEMS DESIGN

AL SYSTEMS DESIGN

f the systems development process
e true nature of the problem specified
d in gaining an overall understanding
stem. This was accomplished through
the preliminary survey required the
ersonnel and to collect and review
d phase problem definition and
tigation and analysis of the current
is portion of the systems project, and
e systems analysis tools and
ve been collected and systematically
d in detail.

From this analysis, alternative plans
problems found in the current system.
ble to begin addressing specific
When? By whom? Where? In what
How can it be collected? In short,
all user requirements for
must be aware of any additional
ed new system. These requirements
demands on the system other than
tem requirements often include, but
erations:

.

ancy.

ving determined user requirements and
design a new system which makes the
urces (personnel, money, machiens,
be aware of what resources are
alysis of the current system. While
t use of these resources, the uses to
nal because many of the activities in
ame resources. The limitations placed
straints. Any constraints that will
have been determined during detailed

a. The process of systems design is concerned with meeting user and system requirements by using the available resources within the constraints imposed by management. Systems design can be thought of as the drawing, planning, and sketching or arranging of many separate elements into a viable whole. While the detailed systems analysis was concerned with WHAT the system is doing, and what it SHOULD BE DOING to meet user requirements, the conceptual systems design phase is concerned with HOW the proposed system will meet these requirements. The design activity which takes place in this phase is at a broad systems design level. The end product of this phase is a set of conceptual specifications which will be the blueprints used in the detailed design phase which follows. The conceptual specifications when completed represent a logical and conceptual design package which contains the items listed below.

(1) Output specifications.

(2) Input specifications.

(3) File specifications.

(4) Decision logic tables.

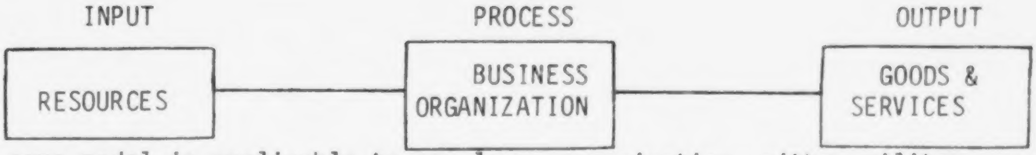
(5) Processing rules.

b. As a result of detailed analysis, the study team should be able to specify what the new system must do, how well the new system must perform, and what the new system must provide for future growth. An equation for a new system may well be written:

NEW SYSTEM = (BEST OF OLD SYSTEM) + (PROBLEM SOLUTIONS) +
(NEW REQIREMENT CAPABILITIES)

c. In order to design the new system, the study team must be thoroughly and totally familiar with the old system. The team's understanding of the organization must be complete. Their knowledge at this stage should allow them to converse with top management concerning the overall mission of the organization and its objectives, and, at the same time, to talk with operational personnel concerning the specific number of characters comprising a data element on a specific organizational document. The application of the systems analysis tools and techniques is what makes this degree of understanding possible.

3.4. ORGANIZATIONAL FUNCTIONS. Previously, a business organization was cited as an example of a system. The business "system" received ''Puts (personnel, material, machines, money) from the environment and converted them into an organization which, in turn, produced goods and services which were then exported back to the environment.


INPUT
PROCESS
OUTPUT
RESOURCES
BUSINESS ORGANIZATION
GOODS & SERVICES

This same model is applicable to any large organization, either military or civilian, which produces either goods or services. The reason for the existence of any organization is the efficient production of its output. Unless it accomplishes its purpose or mission in an efficient and cost-effective manner, it will cease to exist. In organizations, the way to achieve efficient and effective operations has been to resort to functional specialization, that is, specific organizational segments performing necessary organizational functions. For example, in a typical business organization, the following functions are likely to be found: production, inventory control, marketing, finance, personnel, research and development, etc. With minor adjustments for the differences in their basic mission a similar functional structure is found in military organizations.

a. The extremely difficult task faced by the study team during the detailed analysis is to gain a comprehensive understanding of a very complex system (in the case of the total organization) or subsystem (if a subelement is involved). Starting with a general knowledge of the organizations's mission, the team proceeds to a minute level of detail through the use of analysis tools. Figure 3-1 illustrates the uses to which the various tools are put.

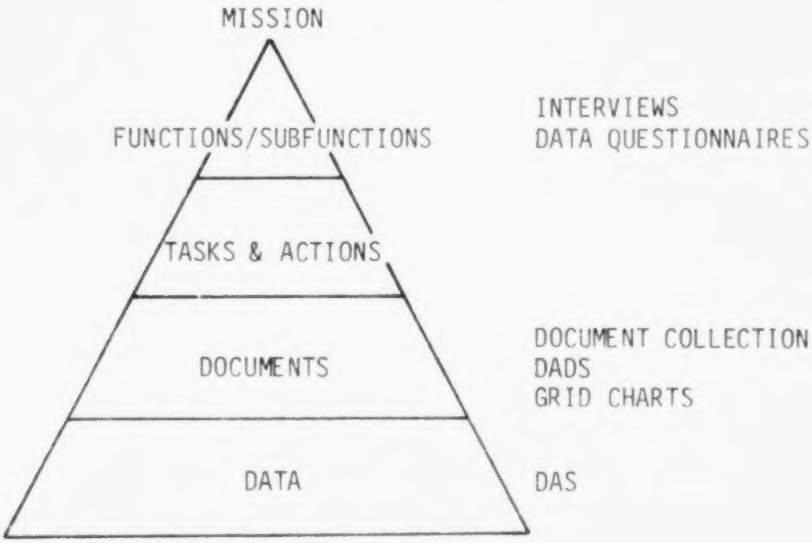

MISSION
FUNCTIONS/SUBFUNCTIONS
INTERVIEWS
DATA QUESTIONNAIRES
TASKS & ACTIONS
DOCUMENTS
DOCUMENT COLLECTION
DADS
GRID CHARTS
DATA
DAS

Figure 3-1.

b. Application of the systems approach is evident once again in the design process. In the conceptual design process, the organizational element under study is viewed as a total system that performs a single function. This function is examined and broken down into separate subfunctions. The subfunctions should be further subdivided and refined to a level of detail that supports general or conceptual systems design. By starting with a single function and progressing to greater levels of detail, the study team or systems analyst progresses from general knowledge to very detailed and specific knowledge of the function being studied.

c. In addition to arriving at very, very detailed functional descriptions, the analyst is able to describe the functions or subfunctions in terms of the processes that occur and identify the necessary inputs and resultant outputs. As a result of this process, each function, subfunction, or sub-subfunction can be represented graphically as an input-process-output model (figure 3-2).

(1) Input. The inputs to functions are the documents, files, records, fields, blocks, etc.

(2) Process. The processing steps (logical decisions and mathematical calculations that support the functions being described.

(3) Output. The end products of the process (documents, reports, files, records, etc.).

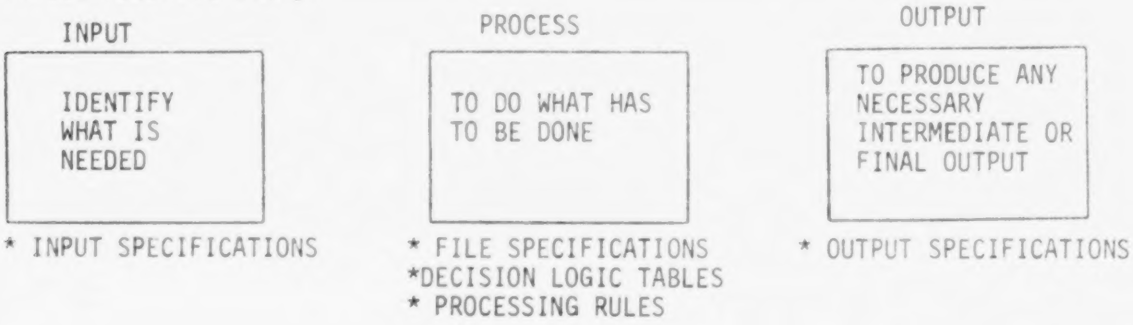

INPUT
IDENTIFY WHAT IS NEEDED
* INPUT SPECIFICATIONS
PROCESS
TO DO WHAT HAS TO BE DONE
* FILE SPECIFICATIONS
*DECISION LOGIC TABLES
* PROCESSING RULES
OUTPUT
TO PRODUCE ANY NECESSARY INTERMEDIATE OR FINAL OUTPUT
* OUTPUT SPECIFICATIONS

Figure 3-2. Input-Process-Output Graphic Relationships.

3.5. CONCEPTUAL SYSTEMS DESIGN PACKAGE.

a. Output Specifications. Through interviews and discussion with users of the system, the systems analyst is able to determine what is needed as output from the system to meet the needs of the user. The design of a new system begins with the design of the system outputs. It is difficult, if not impossible to design the input to a system or to define the processing which is to occur within a system if the desired outputs are not first determined.

In preparing output specifications, the analyst should concentrate only on content, ignoring the output media. Using the records of interviews and the DADS prepared during the detailed analysis phase, the analyst is well aware of the output requirements for the current system. Using these documents, and having discussed with the user the problems in the old system and his new requirements and desires, the analyst is able to accomplish the redesign and consolidation of existing reports and the development of new reports. The development of output specifications concentrates only on content and ignores the media. The actual format and media are determined later during detailed design.

b. Input Specifications. After the contents of the system outputs have been determined, the analyst proceeds to design the system inputs. In developing input specifications the analyst is defining the data that must be brought into the system in order to produce the required output. During the process of developing input specifications the analyst will find that the data needed to produce a given field or data element in an output may be obtained from a number of sources, as shown below:

(1) Data that enters the system on source documents.

(2) Data that is derived or generated as a result of calculations within the computer program.

(3) Data that will likely be available from files or tables within the system and serve as inputs.

c. File Specifications. Probably one of the most important steps in conceptual design is the development of file specifications, for it is through the use of files, both master and intermediate, that data flows through the system, from source input to final output. As mentioned earlier, source documents are often edited and converted to a more rapidly processable medium such as tape or disk. The result is an intermediate file. Often, in a computer run, data derived as a result of calculations is stored for processing in a later computer run; once again making it necessary to develop file specifications. The retention of data for long periods of time, or the need to provide summary data at fixed intervals (weekly, monthly, quarterly, etc.) causes the need for files. As pointed out in Matrix Charting, information that is fixed in nature will be retained on master files. In most systems these files are not used directly to create output reports. Rather, intermediate files are created which contain all of the information needed to produce the reports within a system. The information found on these intermediate files not only is extracted from master files but also results from processing of transaction files (containing source data) and calculations which were performed within programs. These file specifications are media independent. Media and format are not determined until later in the detailed design phase.

d. Decision Logic Tables. The flowchart of the present system provided an overview of the decision logic associated with the flow of data through the system. The decision symbol on the flowcharts performed this function. However, symbols do not provide enough detail to enable a programmer to code the detailed logic associated with each decision in a computer program. The use of symbols for this task would produce a flowchart that would be so complex and confusing as to be worthless in complex decision situations. In these situations the decision logic table is an invaluable tool. A part of its value lies in the fact that it permits the extremely complex logical decisions which must be made within a computer program to be shown graphically. The decision table is used to represent alternative actions based upon conditions which may arise. Therefore, CONDITIONS which must be tested and ACTIONS which are to be taken must be specified. The results of both conditions and actions are called DECISION RULES. Each combination of condition which can occur and the associated action is called a rule.

The decision table, while it may appear complicated, is quite simple. It is based upon an "if. . .then" relationship between conditions and actions. S)mply stated, IF a given set of conditions exists, THEN the following action will be taken. A detailed explanation of decision tables was presented earlier.

e. Processing rules. After the output/input specifications, file specifications, and decision logic tables have been prepared, the analyst is ready to determine, at a broad conceptual level, how the system must operate to produce the required outputs. He must consider all processing from creation of master and intermediate files to preparation of final output reports. The processing that occurs will normally encompass the following items.

(1) Machine edit and validation.

(2) Creation of files (master and intermediate).

(3) Sort, match, merge, and collate with other files.

(4) Extract data from files.

(5) Update files.

(6) Perform calculations and logic decision.

(7) Produce intermediate and final outputs.

The basic tool used to accomplish design of the processing within the system is the flowchart. At this stage, the analyst begins the development of the flowcharts that depict the processing that is to occur in the new system. These flowcharts are "rough", and simply serve to aid the analyst in developing the concept of the new system. Later, in the detailed design phase these rough flowcharts will serve as the basis for the flowchart of the new system.

3.6. AN APPROACH TO CONCEPTUAL SYSTEMS DESIGN. The conceptual systems design package was presented as an accomplished fact. No mention was made of the specific steps or activities involved in the generation or production of the conceptual systems design package. The procedures described below show the process whereby each function, subfunction, sub-subfunction, etc., is elaborated into a proposed new system.

a. Identify all outputs associated with each function or subfunction.

b. List the specific information fields required to prepare that output (develop output specifications).

c. Identify all inputs required to support the generation of the required outputs.

d. List the specific information fields required to produce the

e. Describe the data processing operations, particularly the logical decisions and calculations which must be applied to the data to produce the desired information (develop decision logic tables and processing rules).

f. Identify those elements which can be input once and stored for use in subsequent processing (develop master files).

g. Continue executing steps a through e for each output requirement on a priority basis until all outputs are considered.

h. Develop the data base (all files, master and intermediate) that will support the system most effectively, by considering user requirements, system requirements, data processing methods, and commonality of data (develop file specifications).

i. Based upon developmental constraints, support priorities, and an estimate of development costs, eliminate extreme input, output, and processing specifications.

j. Define the various control points necessary to regulate the data processing activities and insure overall processing quality.

k. Finalize only those output and input specifications which will best satisfy the current systems design.

3.7. CONCEPTUAL DESIGN AND THE SYSTEMS ANALYSIS TOOLS. Many of the systems analysis tools presented earlier are employed during conceptual systems design. The tools are used for fact-finding and fact analysis but they also serve as effective sources of documentation for the proposed new system. The principal tools used during conceptual systems design are the Document Analysis and Data Sheet (DADS), the Data Analysis Sheet (DAS), and the grid chart.

a. All specifications, outputs, inputs, and files should be recorded in such a manner as to facilitate the actual physical design which occurs later. The back side of the DADS can be used for these specifications since it contains spaces where an explicit definition of each data element can be entered. At this time, the data codes portion of the DADS can be used, because the specifications can be coded as follows:

(1) Output specifications - Code 3 (Final Output)

(2) Input specifications - Code 0 (Source inputs)

(3) File specifications - Code 1 (Master Files) and Code 2 (Inter-mediate files).

b. In the analysis of the present system, the DADS was used to prepare both the grid chart and the data analysis sheet (DAS); the same procedure is followed at this time. The procedures to follow in "firming up" the conceptual design are as follows.

(1) List all data elements needed on each final output needed by users to satisfy their information requirements.

(2) Make drafts of all new or revised outputs (code 3 documents only). Contact the user and negotiate the content. Obtain his concurrence and get him to either initial the draft or sign a memo signifying his concurrence.

(3) Prepare a DADS for each new or revised document.

(4) Use the completed DADS to fill out the Output side of a grid chart for all system outputs (code 3 only).

(5) Fill out the Output side of a data analysis sheet (DAS) from the new group of completed DADS. Enter data codes for as many of the data elements as possible.

3.8. DETERMINING I/O DAS. The entering of data codes for the data elements contained on final outputs is an important step in the design process since these codes will aid in the identification of input specifications and file specifications. Once again, the data codes fall into two broad categories, fixed and variable.

FIXED	VARIABLE
I - Identifying	R - Reported
Q - Quantitative	G - Generated
	C - Constants

a. Fixed data elements are usually placed on some type of master file once they are entered into the system. Therefore, it can be said that the data elements coded I and Q lend themselves to retention on master files. Variable data elements are either continually entered into the system on source inputs (R-Reported), or as program constants (C-Constant), or they are generated within the program as the result of calculations (G-Generated).

b. To better illustrate the use of data codes, assume that a user in a personnel application has requested a payroll report. After negotiation, it has been decided that the report should be formatted as follows:

PAYROLL REPORT

EMPLOYEE NUMBER	EMPLOYEE NAME	HOURS WORKED	HOURLY RATE	GROSS PAY
76240	William L. Ditka	40	$6.00	$240.00

A DADS is prepared and then used to produce a DAS. On the DAS, the fields would have the appropriate data codes entered.

DESCRIPTION	VALUE	DATA CODE
EMPLOYEE NUMBER	76240	I
EMPLOYEE NAME	WILLIAM L. DITKA	I
HOURS WORKED	40	R
HOURLY RATE	$6.00	Q
GROSS PAY	$240.00	G

As stated earlier, fixed data elements lend themselves to inclusion on master files. It is obvious that in this system there will be a Master Employee File containing Employee Name and Employee Number as well as some other key fields. Hours Worked is coded R (Reported). R-coded data enters the system via source documents. This indicates that some type of employee time card will have to be designed to bring the data element "Hours Worked" into the system. Hourly Rate is coded Q (Quantitative) and is fixed in nature, meaning that it should be included on a master record or file. Hourly Rate could be included as a data element on the Master Employee File, or a separate Hourly Rate Table might be developed. (Gross Pay would be coded G (Generated) since it is the result of a calculation (e.g., Hours Worked X Hourly Rate).

After filling in the output side of the DAS, the analyst proceeds to fill in the input side. The resulting DAS contains only Final Output (code 3) documents and Source (code 0) documents. For this reason, this DAS is called an INPUTS and OUTPUTS DAS (I/O DAS), and the series of actions that have been accomplished are termed "determining I/O activities".

3.9. DETERMINING FILE SPECIFICATIONS AND PROCESSING RULES. Referring back to the Input-Process-Output relationship shown in Figure 3-2, it becomes apparent that the analyst, in completing the I/O DAS, has made significant progress in his conceptual design efforts. First, he has identified all final outputs of the system, and secondly, he has identified what inputs are needed to produce those outputs. The next step is to determine what has to be done to move the data from source input to final output. Specifically, the PROCESS of the Input-Process-Output model must be determined.

a. The first step in determining the system processing is to develop the file specifications for the system. Files will fall into two basic categories - master files and intermediate files. Master files are permanent in nature, and contain information reflecting the current status of the system. In the example of a payroll system (figure 3-3), the Master Payroll File would contain the employee name, employee number, and other permanent information about the employee such as year-to-date earnings, year-to-date taxes, etc. One of the key functions performed by intermediate files is the periodic update of master files.

b. Intermediate files often contain records of individual transactions which reflect the day-to-day activities of an organization. In the payroll system example, the Master Payroll File would contain year-to-date earnings and would be updated by an intermediate file (Payroll Transaction File) containing the current earnings for the week. The result of processing the intermediate file against the master file would be a new, updated, master file containing new year-to-date earnings, new year-to-date taxes, etc. A payroll data file may also be produced which can be used later as input to a complete run to produce payroll reports.

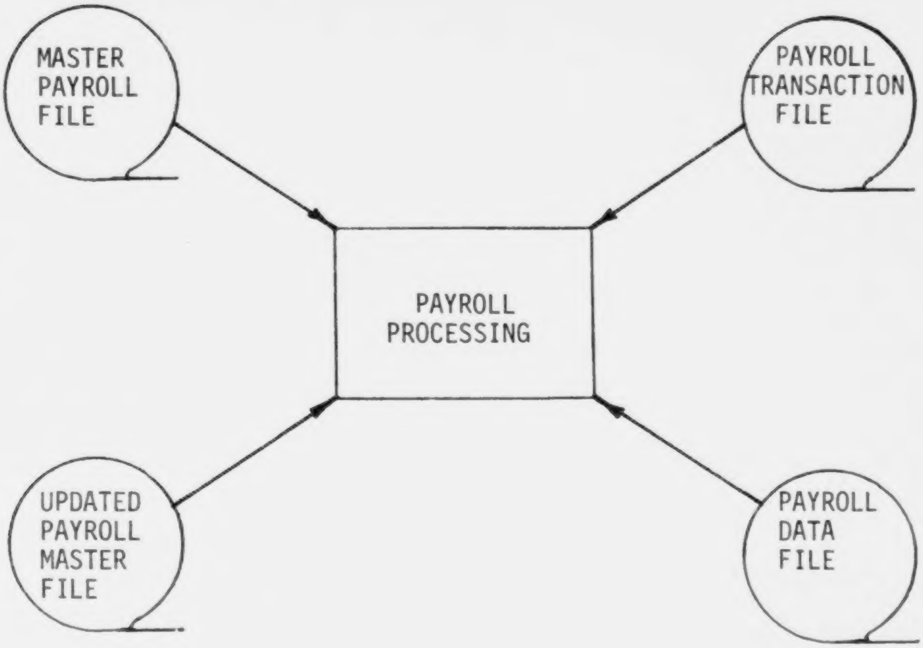

MASTER PAYROLL FILE
PAYROLL TRANSACTION FILE
PAYROLL PROCESSING
UPDATED PAYROLL MASTER FILE
PAYROLL DATA FILE

Figure 3-3.

c. As the analyst wrestles with the problem of how to take the data elements from the source inputs to the final outputs, file specifications for master and intermediate files are developed. Concurrent with the development of files is the development of processing rules, since, in the use and manipulation of files, certain activities must be performed.

(1) Machine edit and validation of source documents.

(2) Creation of master and intermediate files.

(3) Sort, match, merge, and collate with other files.

(4) Extraction of data from files.

(5) Updating of files.

d. When completed, the file specifications are placed on DADS. These DADS are the ones used to update the partially completed grid chart and the partially completed I/O DAS.

3.10. DECISION LOGIC. The point has been made many times that the systems analysis and design process is repetitive and cyclical in nature. The truth of this statement becomes evident once more. While the development of file specifications and processing rules would appear to be a distinct series of activities completely separated from the development of decision logic, this is not the case. Decision logic tables must be developed along with file specifications and processing rules, and they present clearly and precisely the conditions which must be tested and the actions which must be taken within a computer program.

3.11. RELATIONSHIP BETWEEN CONCEPTUAL DESIGN AND DETAILED DESIGN. At the end of conceptual systems design, the analyst should have a general concept of the new system. Armed with the conceptual systems design package and the rough flowcharts, he is better prepared to proceed with detailed design. During the detailed design phase, the analyst will convert the conceptual specifications into a form that can be processed by the computer economically and effectively. The process of detailed design includes these actions.

a. Formatting and layout of inputs and outputs.

b. Selection of a file organization and processing scheme.

c. Design of the systems files.

d. Determination of input, output, and file media.

e. Incorporation of controls to insure processing reliability.

f. Developing the flowchart of the new system.

The resulting detailed systems design includes specifications for programming and files conversion, as well as procedures for data control and computer processing.

Section 2. INPUT/OUTPUT DESIGN

3.12. OUTPUT MEDIA AND FORMS DESIGN. Garbage In - Garbage Out (GIGO) is a commonly used phrase. The analyst would like to see the meaning of GIGO be changed to Good Input - Good Output. However, even with good input, there is no guarantee of good output. The media must be appropriate and the design of the forms used must present the data in the proper manner.

a. Over the last few years, the increased volume of output has outstripped management's ability of assimilating the information presented. With this in mind, the following objectives should be considered.

(1) Reduce the volume of output, especially printed output, to the minimum consistent with achieving the goals of the system.

(2) Arrange the data so that it is easy to understand.

(3) Provide the output to the person who needs it, when he needs it.

b. To accomplish these goals, there are several forms design steps to follow. They are listed below.

(1) Establish the necessity of designing or redesigning forms.

(2) Analyze similar forms not only in your own area, but in adjacent areas as well.

(3) Study the equipment that will be used to prepare the output.

(4) List all types of information to be recorded, and the number of positions which will be allotted for each element to be shown on the output.

(5) Eliminate all unessential information.

(6) Sketch out your proposed form on a lined sheet of paper.

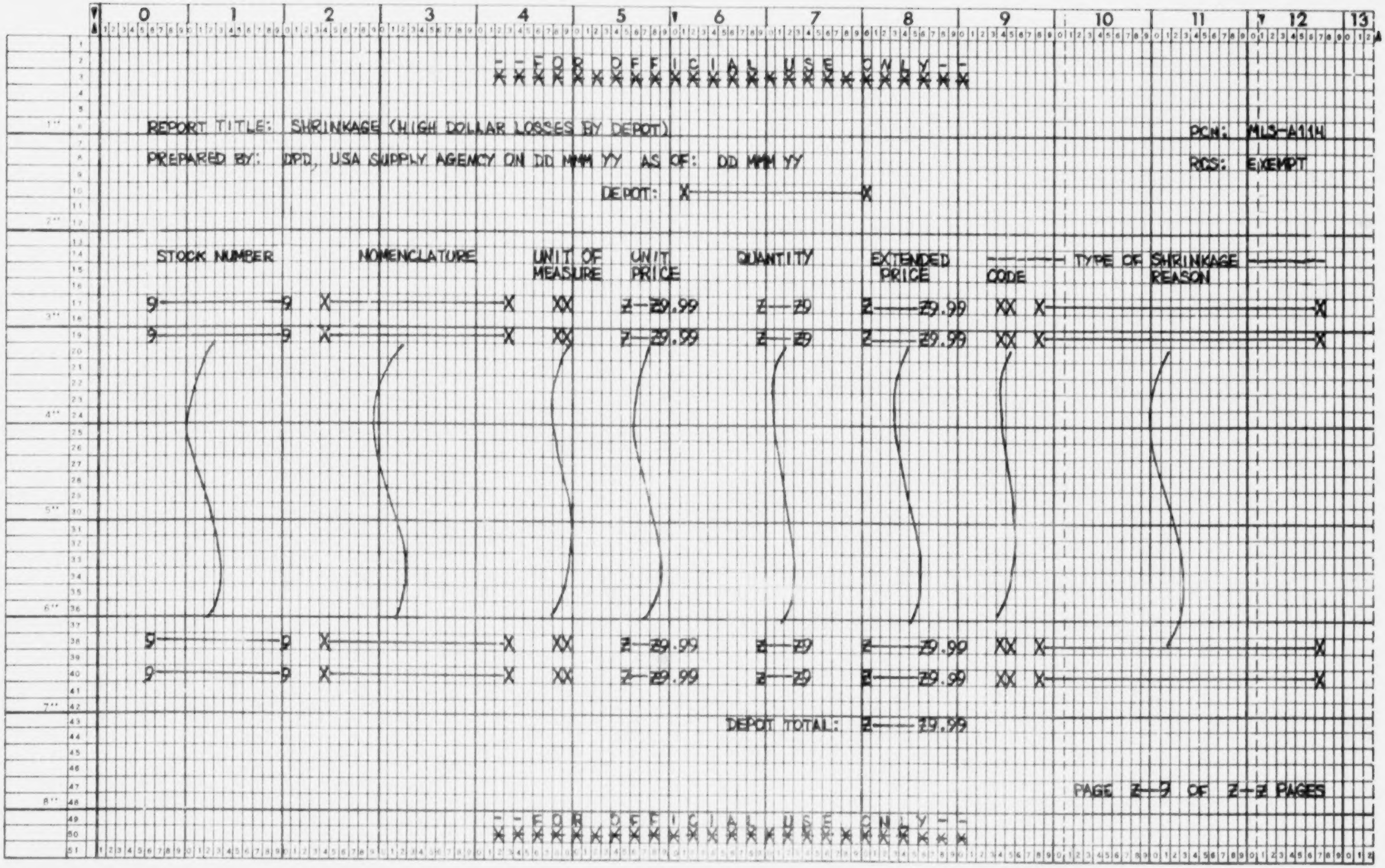
(7) Transfer the final design to a Form Layout Sheet (Figure 3-4).

(8) Obtain final user approval.

Other systems analysis tools will help in accomplishing most of these steps. Such tools as document interviews, DADS, and DAS help determine the need for redesign, the information to be included, and tahe unessential information to be omitted.

c. To determine the validity of a report, its recipient should be asked the following questions.

FORM LAYOUT



```
                    - - F O R   O F F I C I A L   U S E   O N L Y - -
                    X X X X X X X X X X X X X X X X X X X X X X X X X

REPORT TITLE:  SHRINKAGE (HIGH DOLLAR LOSSES BY DEPOT)                           PCN:  M13-A11H
PREPARED BY:  DPD, USA SUPPLY AGENCY ON DD MMM YY  AS OF:  DD MMM YY             RCS:  EXEMPT
                                 DEPOT:  X-----------X

STOCK NUMBER        NOMENCLATURE       UNIT OF  UNIT        QUANTITY   EXTENDED    ----- TYPE OF SHRINKAGE -----
                                       MEASURE  PRICE                  PRICE       CODE       REASON
9---------9  X--------------X          XX       Z-Z9.99     Z-Z9       Z-----Z9.99 XX  X------------------------X
9---------9  X--------------X          XX       Z-Z9.99     Z-Z9       Z-----Z9.99 XX  X------------------------X

9---------9  X--------------X          XX       Z-Z9.99     Z-Z9       Z-----Z9.99 XX  X------------------------X
9---------9  X--------------X          XX       Z-Z9.99     Z-Z9       Z-----Z9.99 XX  X------------------------X
                                                   DEPOT TOTAL:  Z-----Z9.99

                                                                       PAGE  Z-9  OF  Z-Z PAGES
                    - - F O R   O F F I C I A L   U S E   O N L Y - -
                    X X X X X X X X X X X X X X X X X X X X X X X X X
```

ADMINCEN FORM 7027, SEP 78
(ATSG-CSD)

FOR INSTRUCTIONAL PURPOSES ONLY

Figure 3-4.

(1) How many other persons use the report?

(2) How essential is the report to the work of the unit?

(3) How often does the user refer to his copy of the report?

(4) How much of the information on the report is not used bythe interviewee?

(5) Is the data on the report necessary for:

(a) making decisions to take action?

(b) keeping one informed about current conditions?

(c) checking accuracy of other matters?

(6) What would be the effect on work if the user:

(a) did not receive the report at all?

(b) received the report less frequently.

(c) received less information than at present?

(d) received more information than at present?

(7) The cost of preparing this report has been estimated at XX dollars. Is the use of the data considered as justification of this expense?

(8) What other reports, records, or forms are prepared from data on this report?

(9) Can the data on this report be obtained from any other source?

(10) Is the report easy to use and read?

(11) How long is the report retained by the user?

(12) How and where does the user file the report?

(13) How often is the report referred to after its original use?

d. We need to examine the media to be used. Following is a general description of some major output media:

(1) Punched card: The conventional form of this media can contain up to 80 characters of alphabetic, numeric, or special character information.

(2) Cathode Ray Tube (CRT): A CRT displays information on a screen very similar to a television set.

(3) Console Typewriter: A device tied directly to the computer to provide direct, two-way communication between man and machine.

(4) Disk: A storage device on which information is recorded on the magnetized surface of a rotating disk.

(5) Printer: An output device for presenting computer results as numbers, words, or symbols.

(6) Magnetic Tape: A tape or ribbon that is selectively magnetized to represent data.

(7) Mark Sense: A class of techniques that translates marks into machine readable characters.

(8) Paper Tape: A tape made of paper into which holes are punched to represent characters.

(9) Computer Output Microform (COM): One of the newest and potentially most economical forms of output. COM is a technique that will take output directly from a computer tape (which has been substituted for a printer) and produce microfiche or microfilm as a final output.

e. Considering the various media, let us look at advantages and disadvantages of some output techniques. They are shown on this and following pages.

OUTPUT MEDIA	ADVANTAGES	DISADVANTAGES
Card	1. Can be used as input in either the same form or with added information. 2. The media is familiar to most people. 3. Can cut down input errors by pre-punching information and using as input.	1. Slow. 2. Cost of cards. 3. Storage and handling 4. Must be interpreted to be readable by user. 5. Limited record length.
CRT Cathode Ray Tube	1. Very Fast. 2. Easy to change or update.	1. Hard copy not always available. 2. Security of information.

Figure 3-5.

OUTPUT MEDIA	ADVANTAGES	DISADVANTAGES
CRT (continued)	3. Long range cost low.	3. High initial cost. 4. Display size is different from that of most impact printers.
Console Typewriter	1. Communication link between man and machine 2. Good for keeping records of what the computer does. 3. Enables entry of constant data when needed.	1. Not ideal for long reports or messages. 2. Ties up computer during communication.
Disk	1. Speed 2. Cost. 3. Also used as input. 4. High volume of storage 5. Random data access. 6. Virtually unlimited record length.	1. Machine readable only. 2. Transporting data out of facility is difficult.
Printer	1. Faster than cards. 2. User readability. 3. Has the ability to be annotated for future use. 4. Good for historical references.	1. Cost of paper. 2. Storage of reports. 3. Readability of multiple copies.
Magnetic Tape	1. Speed - faster than printer. 2. Cost - reusable media. 3. Also used as input. 4. Large volume of data stored.	1. Machine readable only. 2. Normally sequential files. 3. Slower than disk.

Figure 3-5 continued.

OUTPUT MEDIA	ADVANTAGES	DISADVANTAGES
Magnetic Tape (continued)	5. Faster than cards. 6. Easy shipping. 7. Virtually unlimited record length.	
Mark Sense (using printer)	1. Can take several type forms, card, page,etc. 2. Pre-marking of data for re-input. 3. Increased reliability of input. 4. Speeds input preparation. 5. Faster than cards. 6. Allows for internal sorts and output in order. 7. Room for additional printed characters.	1. Need special forms. 2. Special print chains needed on some machines.
Paper Tape	1. Can get hard copy and machine input at same time. 2. Easier to store than cards. 3. Cost. 4. Unlimited record length	1. Slow preparation and input. 2. Needs converter. 3. Machine readable only. 4. Tears easily. 5. Hard to change data.
COM Computer Output Microform	1. Printing at computer tape speed. 2. Forms are optically printed simultaneously with the data eliminating need for preprinted paper forms.	1. Some reprogramming. 2. Not good for small volumes.

Figure 3-5 continued.

OUTPUT MEDIA	ADVANTAGES	DISADVANTAGES
COM (continued)	3. Material cost reduced no need for paper. 4. Storage savings - only 4% of paper space. 5. Filing cost reduced - fiche contains 269 documents. 6. High quality copies. 7. Virtually noiseless operation. 8. Reduced mailing costs. 9. Paper copies in seconds 10. Fixed record sequence.	

Figure 3-5 (continued).

f. The design of output for most of these media is unique. We will examine the basics of designing a printer-generated report. Some of the basics will apply to other techniques also. In designing your report on the Form Layout Sheet, these tips will be helpful.

(1) Place filing information at the top of the form.

(2) Place analytical figures side-by-side to facilitate comparison by the user.

(3) Place important information near the top of the layout.

(4) Provide room for legends, if required, to aid in understanding the form.

(5) Always uniquely title a form and include a form number (RCS or RIN, see AR 335-15; PCN, see TB 18-103), date, and page number.

(6) Avoid horizontal and vertical rulings as much as possible to eliminate the need for printer adjustments.

(7) If possible, choose a standard form. In this regard, it should be noted that nearly all DPA's have a standard report titling format to be followed by the analyst.

(8) Make sure the form length is compatible with the spacing used and the printer carriage control tape.

(9) Eliminate as much unnecessary information as possible.

(10) How many copies are needed? Should carbon copies be produced or can they be made by electrostatic copiers?

(11) Include security classification at top and bottom of each page if applicable; or Privacy Act indication, if applicable.

(12) Make repetitive information part of header or column titles, if possible.

(13) Arrange data so that it presents information progressing from general to specific (from left to right).

(14) Consider the manner in which the report will be bound when laying it out. Leave enough space on the side being bound so that data will not be obscured by the binding.

(15) Include appropriate column headers.

g. For preprinted forms, also consider the following items:

(1) Determine what data is to be preprinted on the form, and what additional data will be printed by your equipment. Consider:

(a) Commander's desires.

(b) Cost.

(c) Purpose of output.

(2) Use bold type to emphasize special information, so that titles or headings standout.

(3) Use double ruled lines to set off sections, but keep in mind the problems of printer alignment.

(4) Consider guide marks for printer alignment or folding.

(5) Use some kind of protection for payroll and other checks.

b. Even with all the different types of output available, the most frequently used is still thje printer. In designing printer output, a form layout sheet, sometimes called a spacing chart, is used. Figure 3-4 is an example of a completed form layout sheet. On this sheet, each numbered block represents one printable position. It is filled out by indicating what should be printed by placing titles, headings, and data fields where desired. Shown below are the most frequently used field identification symbols.

A - Alphabetic characters
9 - Numeric characters
X - Alphanumeric characters
Z - Zero suppression in numeric field

Using these symbols and the guidance provided in this text, you are ready to complete a form layout. Remember that prior to adopting a new output, final user approval is required.

As an analyst, the starting point for designing inputs and outputs for a new system is on the output side. Once this is completed, work can begin on the design of the system's inputs.

3-13. INPUT MEDIA AND FORMS DESIGN. Accurate input is essential. In the discussion of outputs, it was noted that accurate input did not guarantee accurate output. However, good output cannot be obtained without good input. A great deal of a system's run time is devoted to processing input. Further, as much as half the cost of operating a computer system is in the collection and preparation of data for processing.

a. There are some basic input objectives to keep in mind. These are as follows:

(1) Whenever possible, capture the data in a machine readable form at the point where the transaction occurs. This practice is called Source Data Automation (SDA).

(2) Design the input so that it can be easily used.

(3) Capture the input when it is needed.

(4) Reduce the total volume of input to the lowest practical level.

b. To accomplish these goals there are several steps to follow that are similar to those used in output design.

(1) Establish the necessity of designing or redesigning a form.

(2) Analyze similar forms in both your own area of the system and in adjacent areas.

(3) Study the characteristics and limitations of the equipment to be used for input.

(4) List all types of information that must be recorded and the number of positions which will be required for each item of data to be input. Also, indicate if the data is alphabetic, numeric, or alphanumeric.

(5) Eliminate all unessential information on the form.

(6) Sketch out your proposed form.

(7) Transfer the final design to a layout sheet for the particular media.

(8) Get final user approval in writing.

c. There are systems analysis tools that will help to accomplish most of these steps. Such tools as the document interview, the DADS, and the DAS will help in determining the need for input redesign and will indicate the information that should be omitted.

d. Even though we normally think of input as an automated product, analysts do get involved in the design of non-automated inputs. In these cases, the analyst should remember the following hints:

(1) The document should read from left to right and from top to bottom.

(2) Data fields should be grouped in logical or natural sequence.

(3) If data fields are to be typed (as with OCR), the form should line up properly.

(4) White spaces should be used to make the data conversion job easier.

(5) Enough room should be provided in each data field so that the form can be filled out properly.

e. Examine Figure 3-6 and determine if it agrees with these hints. Is it a good form or a poor form? What is wrong with it? You probably found at least the following things wrong, and possibly others as well.

(1) The form does not read from left to right in a natural sequence. For example, age and phone number are out of place.

(2) Fields are not grouped logically. Birthplace and birthdate are separated by two other fields.

(3) Several items are very close together and they are not in a good sequence for data conversion. The operator would have to skip all over the form to get the final data in the proper sequence.

(4) There is too much room in several places. For example, birthplace has a long line when a two-digit response is all that is needed.

(5) There is no title on the form.

(6) The form is not self-explanatory. What is wanted at "year"? Birthplace is a two-position field. Is place of birth desired?

(7) Are three positions needed for age?

(8) There is not enough space for name and address.

(9) Are both birthdate and age needed?

PLEASE PRINT CLEARLY

NAME ______________________ (1-15)   AGE __________ (48-50)

ADDRESS ______________________ (16-40)   PHONE __________ (41-47)

BIRTHPLACE ______________________ (51-52)

EDUCATION LEVEL ______________________ (59-60)

YEAR ______________________ (61-62)

BIRTHDATE ______________________ (53-58)

Figure 3-6.

f. Is anything right with this form? It does indicate the columns to be punched, even if their organization leaves much to be desired. Now, examine Figure 3.7.

BIOGRAPHICAL DATA

NAME ____________________(1-20)

STREET
ADDRESS _______________(21-35)

CITY _______________(36-50)

STATE __(51-52) ZIP CODE _____(53-57)

HOME PHONE ___ ____ (58-64)

BIRTHDATE (Numeric day-month-year) __ - __ - __ (65-70)

STATE OF BIRTH __ (71-72)

Highest Education	High School	1	Masters	3	
Level Completed (Check 1)	4 yr college	2	PHD	4	(73)

Year Education Completed __ (74-75)

Figure 3-7.

After having looked at this form, you will note that -

(1) The form reads basically from top to bottom and from left to right.

(2) Data fields are logically grouped and spaces are used to separate groups of data.

(3) On this form, the user is guided to print and to limit himself to the desired number of spaces.

(4) The form is self-explanatory.

(5) Are both age and date of birth needed? Possibly not.

(6) Data reduction order follows from top to bottom.

(7) Even though Figure 3-7 is not a perfect form, it is far better than Figure 3-6.

g. Some of the media that were not defined as outputs are:

(1) Key-to-tape or Key-to-disk. Both types replace the end product of keypunching (cards) with tape or disk. Keep in mind that most cards are only keypunched so that they can be read to tape or disk for later processing.

(2) Magnetic Ink Character Recognition (MICR). This technique uses a standard character set that is machine readable - often used by the banking community on checks.

(3) Optical Character Recognition (OCR). This technique reads characters from a limited set.

With the media in mind, let us consider the advantages and disadvantages of each type. They are shown in Figure 3-8.

INPUT MEDIA	ADVANTAGES	DISADVANTAGES
Card	1. Familiar media. 2. Ease of use.	1. Time and cost to produce. 2. Storage space and environmental requirements. 3. Error Correction. 4. Verification is an additional step. 5. Security problems.
Console Typewriter	1. Communication link between man and machine. 2. Enables entry of constant information when needed.	1. Inappropriate for large volumes of input. 2. Ties up the computer.
CRT Cathode Ray Tube	1. Speed for low volume. 2. Low Long range cost.	1. No hard copy of input. 2. Security. 3. High Initial Cost
Key-to-tape Key-to-disk	1. Increased productivity. 2. Media cost - reusable tape or disk vs one-time card. 3. File integrity.	1. 1-time retraining cost. 2. 1-time reprogramming cost 3. Limited use in a small batch environment.

Figure 3-8.

INPUT MEDIA	ADVANTAGES	DISADVANTAGES
Key-to-tape Key-to-disk (continued)	4. Input speed into computer. 5. Low machine cost. 6. Error correction capability. 7. Input format flexibility. 8. Ability to handle large volumes of data.	
Mark Sense	1. Easy to use. 2. Reduces manual steps. 3. Increased Reliability.	1. Special forms required. 2. Repeated handling can cause machine malfunction.
MICR Magnetic Ink Character Recognition	1. Relatively fast and reliable. 2. Eliminates several manual steps.	1. Highly specialized character set. 2. High accuracy required.
OCR Optical Character Recognition	1. Eliminates several manual steps. 2. Cheaper than keypunching or keytaping.	1. Requires very high accuracy and quality. 2. High Reject rate. 3. Limited Character range.
Paper tape	1. Communication between man and machine or between machines. 2. No limit to record size. 3. Machine output can be used as computer input.	1. Not user readable. 2. Difficult to change. 3. Limited uses.

Figure 3-8 continued.

h. The question, "Does the existing input need to be revised or replaced?" is a difficult one. The following considerations will make it easier to answer.

(1) Is the present input form specific and self-explanatory, or is it vague and confusing?

(2) Is the present input document trying to provide too much data for too many purposes?

(3) Is the present input document prepared in the correct number of copies?

(4) How is the document routed to other using functions?

(5) Is the document distributed to too many people?

(6) Are there controls to prevent loss or compromise of data?

(7) Is there a method for finding a missing document?

(8) Is there a method for validating the accuracy of information on the input document?

(9) If the present system is manual, can the input form be used in a proposed automated system?

(10) What is the cost of the present input form? Is there a less expensive method for producing it?

(11) Allow for future expansion or regulatory changes.

(12) Place fields to be sorted in a logical sequence.

(13) If knowledge of file design is available, take this into consideration when designing input documents.

i. If these steps and procedures are followed, is it guaranteed that the results will be a good input? Usually, the answer is yes; however, what constitutes a good input can only be judged by the user. Consequently the final step in the design of inputs or outputs is to get them accepted and approved by the user.

Section 3. MASTER FILES AND MEDIA

3.14. INTRODUCTION. Any data processing system (manual or automated) is supported by files. Files exist in order to support the day-to-day operations. How well the system functions or how well management's objectives are satisfied is a direct reflection on the files supporting the system. Too many times, reports are prepared from files which contain outdated or erroneous information. This breeds mistrust of ADP-produced reports. The files you design must be able to meet the requirements imposed by management. File content, structure, and selection of the media are your responsibility.

3.15. TERMINOLOGY. Basic terms applying to files are as follows:

a. File. A collection of related records treated as a unit. Any collection of informational items similar to one another in purpose, form, and content.

b. Record. A group of related facts or fields of information treated as a unit.

c. Field. A set of one or more characters which is treated as a whole. A specified area of a record used for a particular item of data.

d. Key. A field or group of fields utilized in the unique identification of a record within a file.

e. Data Base. The set of data or information (one or more related files) on which operations and conclusions can be based.

f. File Maintenance. The addition to, and/or deletion of records from, a file. The act of changing an existing key, thereby causing the repositioning of a record within a file.

g. File Update. The changing of any information, other than the record key, stored in a field or fields of an existing record in the system.

3.16. GENERAL FILE CHARACTERISTICS. All files, regardless of the selected media, will have the same general characteristics. Record format and blocking factor tend to be storage media independent when speaking of tape and direct access storage devices.

a. Record Format. Basically, there are two ways a record can be formatted within a file.

(1) Fixed Length Records. Fixed length records are records whose lengths are fixed at a predetermined length throughout the file. Each record in the file contains the same number of fields and each field contains the same number of characters. This is currently the most widely selected format. Because of programming ease and design simplicity, fewer logic steps and instructions are needed. File maintenance and update routines are much easier to design and program. This means that a lesser degree of programmer training and skill is required. Fixed length records are preferred when -

(a) A lack of programmer training or expertise exists.

(b) Design simplicity and programming ease are desired.

(2) Variable Length Records. Variable length records are records whose lengths vary throughout the file. The length is determined by the actual number of data positions needed for each individual record. The exact record length is not predetermined, but a maximum record length must be established. Variable length records may be classified in three basic types.

(a) Type 1 - The record contains a data element or elements that are fixed in length, but the number of repetitions of that data element or elements varies. An example of this would be a payroll file showing deductions. The number of repititions of the field for deductions would vary depending on the number of deductions established for each employee (Figure 3-9).

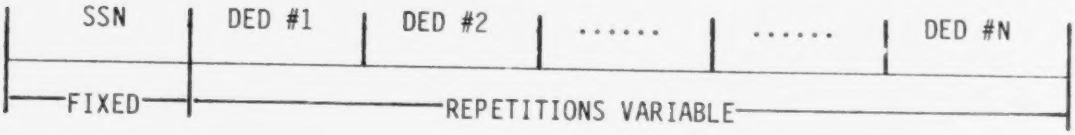


Figure 3-9.

(b) Type 2 - The record contains a fixed number of data ments but one or more data elements may vary in length. An example of this type would be a record which contains a field for storing an individual's name. The length of the field would vary with the number of characters in each person's name (Figure 3-10).

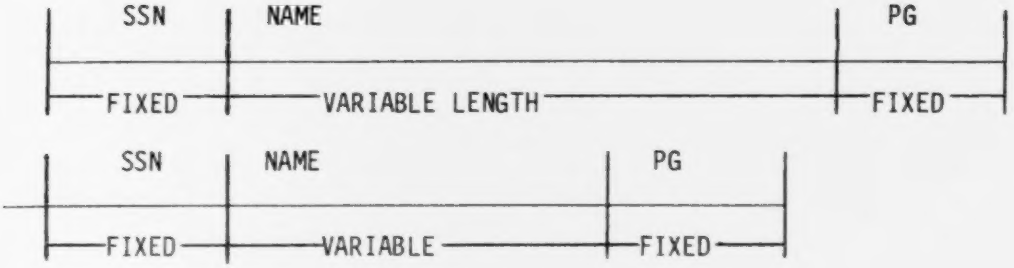

SSN
NAME
PG
FIXED
VARIABLE LENGTH
FIXED
SSN
NAME
PG
FIXED
VARIABLE
FIXED

Figure 3-10.

(c) Type 3 - The file contains records of a fixed format but the file is composed of different types of records. An example of this would be a system where several different types of transactions exist. A file of the transactions is created, thus, the program which performs the file update will not know the type transaction or record length until the record is retrieved (Figure 3-11

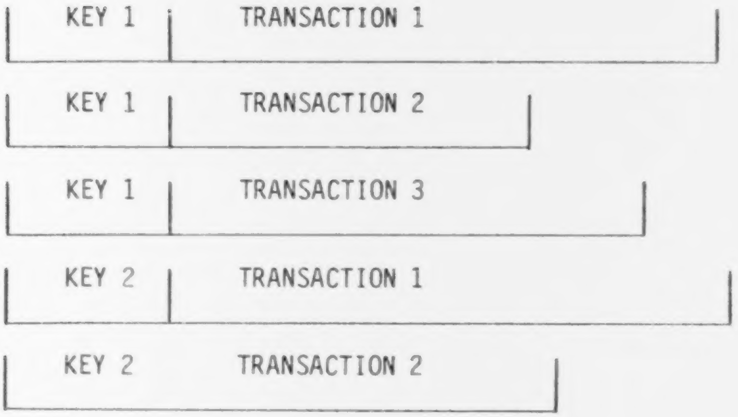

KEY 1
TRANSACTION 1
KEY 1
TRANSACTION 2
KEY 1
TRANSACTION 3
KEY 2
TRANSACTION 1
KEY 2
TRANSACTION 2

Figure 3-11.

(3) Although additional experience and expertise on the part of the programmer will be required, variable length records should be considered as an alternative to fixed length records when any of the following conditions exist -

(a) Considerable variability exists in the record lengths in a file.

(b) The DPI management desires to conserve file space and/or processing time.

(c) Trailer records or linking to a subordinate file are not possible.

(4) Comparison of Fixed Length and Variable Length Files. Consider a file composed of records with the following data characteristics:

SSN:	9 Bytes
NAME:	20 Bytes
GRADE:	2 Bytes
Unit of Asgmt	20 Bytes
Total Record Length	51 Bytes

The logical selection here would be fixed length. However, if the user wanted to keep information on the last 20 assignments, the record would then have 400 bytes for unit of assignment alone.

SSN:	9 Bytes
NAME	20 Bytes
GRADE	2 Bytes
Unit of Asgmt	400 Bytes
Total Record Length	431 Bytes

An individual who has several years in the service may well have 20 assignments and his record may require the mamixum length. But what about the individual just coming into the service? Consideration should now be given to using variable length records. As an individual gains a new assignment, 20 bytes are added to his record until a maximum of 20 assignments is reached. At this time, the record would cease to grow. Using variable length records conserves space by eliminating the storing of blanks when there are less than 20 assignments.

b. Blocking. Physically, records can be stored on media either individually or in a block of two or more records. A file is defined as being "blocked" if there is more than one logical record per physical record, and "unblocked" if there is only one logical record per physical record. Some new terms should not be defined.

(1) Logical Record (LR): A record as previously defined is a logical record; that is, a group of related facts or fields of information treated as a unit. It is the entity, or all of the facts pertaining to a particular event, person, etc.

(2) Physical Record (PR): One or more logical records within a block. The terms block and physical record are synonymous.

(3) Blocking Factor (BF): The number of logical records in a physical record. The size of the physical record will be the product of the blocking factor and the size of one logical record, and is expressed as:

$$PR = LR \times BF$$

When a file is read, one physical record is moved from the input/output (I/O) device to memory. But you, the analyst and programmer, are interested in the logical record and the contents thereof. At the same time you must consider the physical record and the impact it will have on the system. When establishing the file, it may be one of the following, based upon record format and blocking:

Unblocked, Fixed Length	Unblocked, Variable Length
Blocked, Fixed Length	Blocked, Variable Length

3.17. BUFFERING. Each I/O file used by an application program requires an area in main memory where the file is read into or written from. This area is called a buffer. The size of the buffer area is based upon the length of the physical record, because one physical record at a time is transferred between main memory and the I/O device. Therefore, the amount of main memory available has a definite effect on the selected blocking factor or record length of a file. Some systems allow more than one buffer area to be assigned to a file. The standard default under ANS COBOL is double buffering. This fact must also be considered since it further reduces the amount of main memory available to hold a program. There are definite advantages to blocking the buffer and the relationship of the physical record. Using the "READ" instruction in COBOL as an example, the following happens.

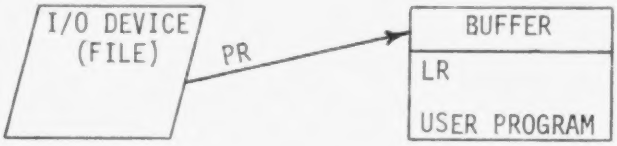


Figure 3-12.

a. Unblocked Files: During each execution of a READ instruction, a physical record is transferred into main memory. In this case a physical record is equal in length to one logical record.

b. Blocked Files (BF = 3).

(1) Single Buffering -

(a) First execution of READ -

Physical record from device to memory (buffer area). First logical record from buffer to user program.

(b) Second Execution of READ -

Second logical record from buffer to user program.

(c) Third execution of READ -

Third logical record from buffer to user program.

(d) Fourth execution of READ -

Physical record from device to memory. First logical record from buffer to user program.

(a) First execution of READ -

First physical record from device to memory (Buffer 1). First logical record from Buffer 1 to user program. Second physical record from device to memory (Buffer 2).

(b) Second execution of READ -

Second logical record from buffer 1 to user program.

(c) Third execution of READ -

Third logical record from buffer 1 to user program.

(d) Fourth execution of READ -

First logical record from buffer 2 to user program. Third physical record from device to buffer 1.

(e) Fifth execution of READ -

Second logical record from buffer 2 to user program.

(f) Sixth execution of READ -

Third logical record from buffer 2 to user program.

(g) Seventh execution of READ -

First logical record from buffer 1 to user program. Next physical record from device to buffer 1.

Blocking the files reduces the number of physical moves of data from the I/O device to main memory. In double buffering, one of the buffers always holds one record ahead on the one we are processing. Both blocking the files and increasing the number of buffers per file decreases the "wait" time for the information. However, both increase main memory requirements. When using double buffering, the buffer size requirement is twice the size of the physical record.

3.18. MAGNETIC TAPE FILE CONCEPTS. One medium on which data can be stored is magnetic tape. Standard magnetic tape in use on computers is 1/2 inch wide, 2400 feet long, with an iron oxide coating. The surface of the tape is divided into tracks, the number of which will vary depending upon equipment generation and manufacturer. For example, third-generation IBM tapes are 9-track, while second-generation IBM tapes are 7-channel. Information is stored by the presence or absence of magnetized bits, forming a character across the tracks of the tape.

Associated with each physical record is an inter-record gap (IRG). The IRG is unused space on tape which allows for start/stop action of the tape drive. When the command to read or write is given, the tape must go from a dead stop to a speed of approximately 112 inches per second. At the conclusion of the operation the tape comes to a complete stop. On the IBM 360, the time required for speed up or slow down results in an IRG of .60 inches.

a. Restrictions for Tape Media. The physical characteristics of tape place certain restrictions on the programmer when processing files on this media.

(1) The use of tape files permits only sequential file structure. Files are maintained in a sequence based on the "key" of the file. Processing of the file begins with the first record of the file, and ends with the last record, having proceeeded through the entire file. Each physical record on the file must be moved from the I/O device to main memory.

(2) The use of tape requires transaction files to be in the same sequence as master files against which they are processed. When a record on the master file has been read, it has passed the read-write head and cannot be accessed again unless the tape is rewound. If, later on in processing, a

transaction comes up for the same master record, the master record cannot be found again unless the tape is restarted from the beginning. Additional problems can arise if there is more than one transaction for the same master record. Multiple records for the same "key" could be created if sufficient care is not exercised in the processing.

(3) Using tape requires the creation of a new master file as the result of every update or file maintenance run. It would be a difficult task at best, if not an impossible one, to write a "new" record over the "old" record on a magnetic tape. Once a tape file has been described as an input file, it cannot be used as an output file. If only one data element in the file is to be changed, then the entire file must be recreated as output.

b. Blocked vs. Unblocked Files. Blocking of records on tape should be considered in designing files. There is no limit to the size of a physical record other than the limit established by the amount of memory available for use as a buffer for the file. If enough memory were available, a physical record could be the entire length of a reel of tape. That, of course, doesn't make sense, but blocking files should be considered.

(1) Blocking of records on a tape file definitely has an impact on the processing time because it causes reductions in -

(a) The number of times data is physically moved from an I/O device to main memory or from main memory to an I/O device.

(b) The number of IRG's.

(c) The physical size of the file.

(2) With blocked files, the total number of physical records is reduced. Associated with each physical record is an interrecord gap (IRG); therefore, the blocked file would reduce the number of IRG's and the physical space required on tape. Since one physical record at a time is moved between the I/O device and main memory, a blocked file would cause more logical records to be moved at one time, and reduce the number of start/stops of the device itself. The following examples point out the effect of blocking a file.

UNBLOCKED vs BLOCKED

(a) Unblocked:

Logical record size: 80 bytes
Tape density in bytes per inch (BPI): 1600
Number of logical records: 20,000
Blocking factor (BF) = 1

Using the above file description, we want to: first, determine the amount of tape required to hold the data; and second, determine the amount of tape required for IRG's.

$$\text{INFORMATION} = \frac{\text{Size LR X Nr PR}}{\text{BPI}}$$

$$= \frac{80 \text{ X } 20{,}000 = 1{,}000 \text{ inches}}{16000}$$

$$\text{IRG} = \text{Size IRG X Nr PR}$$

$$= .60 \text{ X } 20{,}000 = 12{,}000 \text{ inches}$$

Thus, the total file would require 13,000 inches or 1,083.3 feet of tape.

(b) Blocked

Logical record size: 80 bytes
Tape density: 1600 BPI
Number of LR: 20,000
Blocking Factor: 20

The same file description applies, except that the blocking factor has been changed to 20. This reduces the number of physical records and IRG's. The same two calculations as in the preceding example will be made.

$$\text{INFORMATION} - \frac{80 \text{ X } 20{,}000}{1600} = 1{,}000 \text{ in.} \qquad \text{IRG} = .60 \text{ X } 1000 = 600 \text{ in.}$$

In this case, there has been no change in the amount of space required for the data. The entire file in the example above would only require 1,600 inches of tape as compared to the 13,000 inches required for the unblocked file. In example (a), more space would be allocated for IRG's than for data. By changing the blocking factor to 20, as in example (b), the number of physical records is reduced from 20,000 to 1,000, reducing the amount of space required for IRG's to less than that required to store the data. In addition the number of physical start/stops has been reduced. By the same token, the main memory space requirements has increased. Using double buffering, the first example would require a buffer area of only 160 bytes. The second example increases the requirement to 3,200 bytes.

3.19. DIRECT ACCESS STORAGE CONCEPTS. Another medium on which data can be stored is the direct access storage device (DASD), such as disk, drum, etc. Associated with each physical record on these media is a unique address and discrete location. There is no dependency on the positioning of the records. In this case, each physical record is individually addressable; but in a tape file it is not. On a DASD, this allows records to be stored so that a single record can be retrieved without a search of the entire file. Thus, DASD permits retrieval, update, and file maintenance without the time penalty of searching the entire file. This addressability concept is unique to direct access storage devices.

Disk packs used in DASD devices are composed of a number of read/write surfaces. These surfaces are either ferrous oxide or nickel-cobolt over a non-magnetizable metal; i.e., aluminum, magnesium, etc. These plates, or read/write surfaces, are stacked to form a disk pack. The number of plates will vary depending on the type of device. The top of the upper plate and the bottom of the lower plate are not used for storing data, but are for protection.

3.20. BASIC METHODS OF DISK FILE ORGANIZATION. There are three basic file structures associated with direct access storage devices. Each structure has some freedom in file update by not forcing the recreation of a new master file as in tape operations. This addressablity concept is used in all three structures, in some it is more apparent than in others. The three basic structures are sequential, indexed sequential, and direct (random) files.

a. Sequential Organization. The organization of a sequential disk file is the same as a tape file. The file is maintained sequentially based on the selected "key". Therefore, just as in a tape file, the transactions being processed against the file must be in the same sequence. When using a sequentially organized disk file, then, we are more or less using the disk as a high-speed tape. The savings is definitely in the speed of the disk over tape and being able to take advantage of addressability to separate the file maintenance and file update functions.

(1) File Maintenance. Since the file is maintained sequentially and the physical characteristics of the file are being changed by the addition or deletion of records, this structure is handled the same as a tape file. New records are inserted in the proper place based on the key, and deleted records are removed. This type of processing requires the creation of a new master file containing both the retained and the new records, all in the proper sequence. Addressability will not provide any assistance in file maintenance since the addresses will change. The locations of records are determined by the relationship of keys within the file. New records are inserted into the proper places, assigned the proper keys and addresses, and the retained records from the old file are shifted to new locations with new addresses assigned to them.

(2) File Update. Since we are only concerned with changing the data within one record, file update has no effect on the physical characteristics of the file. Consequently, addressability may be advantageous. Since the address of the physical record is a part of the record now stored on disk, the address is moved to main memory along with the physical record. Nothing is done that affects or changes the physical location of the record.

After the record is updated, why not put the record right back into the old file and at the same location? By using the command "REWRITE" we can place the physical record back into the location specified by the address associated with the record (Figure 3-13).

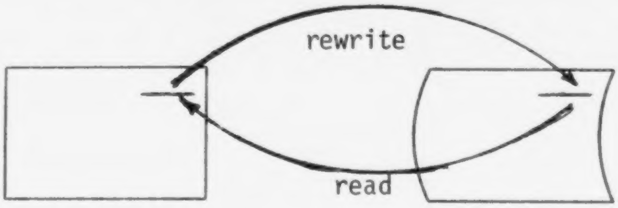


Figure 3-13.

Handling the record this way eliminates the creation of a new master file. The updated record is written over the old record, destroying what was previously recorded at that location. Those which were not affected by transactions remain as they were. There was no "write" action required for them as in a tape file. In a tape file, each record had to be read and written regardless if there was an update or not. Handling files in this way, using the "rewrite" does require some additional effort in the design area to create a backup file. With tape files, there is automatically a backup simply by retaining the old file. But with disk you have no old file. Some time during the processing, arrangements must be made to copy the file, then the copies become your backup (Figure 3-14).

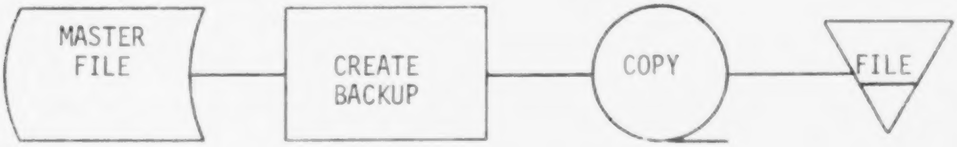


Figure 3-14.

(3) Use of Sequentially Organized Files. Sequential files are used in those cases where the majority of the logical records will be accessed in a given run, or where there exists a high ratio of logical records to transactions. By their very nature and usage within a system, intermediate files and tables lend themselves to sequential organization.

(4) Programming Aspect. From a programmer's point of view, sequential disk files are the easiest to plan, program, and process. The overall logic is much the same as with tape files. The file is processed in a sequential manner, taking each logical record as it comes. Blocked files are very easy to handle. The system will take care of blocking and unblocking the file. As a result, the programmer only has to think of a logical record. Again, remember that both the file being used and the transactions

applied to the file must be in the same sequence. Also, the address for the record is assigned by the system during file maintenance and is of no concern to the programmer. The address will be there when the file is processed, allowing the programmer to take advantage of it through the use of the "Rewrite" command.

b. Direct (Random) Organization. In direct or random organization, data is not stored in sequence based on the key as in a sequential file. Now each physical record in the file is assigned a unique storage address based on some predictable relationship between the key of the record and the address of that record on disk. This file structure makes maximum utilization of the addressability concept in that a sequential search of the file is not performed, but the record is accessed by means of a rapid calculation process or address generation. If the file is carefully organized, rapid access to any record in the file is possible.

(1) File Maintenance/File Update. With this type of structure, there is no need to separate the two functions. In neither case will a new master file be created. For file update, the "Rewrite" command will be used in the same manner as for updating a sequential file. The record will return to its original location. In file maintenance, a new record will be stored at an address generated based on the key. Records to be deleted can simply be flagged, but not removed, until the file is reorganized. Therefore, in both file maintenance and file update, the recreating of a master file is eliminated. Again though, there must be provisions made for producing backup files within the system.

(2) Use of Direct (Random) Organized Files: The direct file organization is used in those cases where the system is transaction or retrieval oriented. Being able to handle transactions as they occur is of primary concern. Since there is no sequence in which the transactions can occur, the system must be capable of handling the transactions in no given sequence, or just as the transactions arrive. This file structure would be used in those systems where the files must reflect the most current information; for example, Point-of-Sales: the transaction updates the balance on hand at the time the sale occurs. There will also be systems with critical response times regarding inquiries made by the user. An example of a critical response time would be the Airlines Reservation System. In either a transaction or retrieval oriented system, there is no sequence to the actions and each action must be handled as it occurs. The system must be able to go directly to the appropriate record without the need for searching a file.

(3) Programming Aspect: Since a relationship exists between the key of a file and the address of the record, an algorithm must be developed. The location of a record is determined by this address generation. The programmer has complete freedom in deciding where the records are to be located; consequently, the logic and programming are his responsibility.

This address generation will then b used during file creation to assign specific locations to records. The same algorithm must be used whenever that file is processed. Care must be exercised in developing the address scheme, because the resulting address must -

(a) Fall within the area reserved for the file on the direct access storage device.

(b) Be distributed evenly across the range of possible addresses so that there are few synonyms, or none at all. A synonym is two or more keys which generate the same address. Any synonym must be accounted for and stored elsewhere. There is no way to completely eliminate synonyms, but there are ways to minimize their occurrence:

1. Allot more space for the file than is actually required to hold the records. As a rule of thumb, allow for approximately 20% additional records. The term "packing factor" relates to the percentage of the reserved space actually being used to store records. In this respect, then, we are speaking of an 80% packing factor.

2. Randomize to track address rather than to record address. By doing this, the next available location on the track will be used, and overflow will not occur until a track is full. Randomizing to a record address will frequently cause overflow or synonyms.

Consideration must be given to blocked or unblocked files. The file may be either; but if the file is blocked, the programmer has the responsibility of blocking or unblocking the file. This is an exception to blocking being performed by the operating system because the system can only handle the physical record. Normal procedure on direct files is to use unblocked files.

c. Indexed Sequential Organization. With an indexed sequential file, you have the best of both worlds. The file can be processed sequentially or, by referencing indexes, the file can be processed randomly. In an indexed sequential file, the records are stored sequentially by key. The difference is the indices which are generated by the system. These indices allow the random accessing of the files. Therefore, the file can either be processed sequentially (ignoring the indices), or randomly using the indices associated with the file. With sequential and direct file organization, there was only one area on the storage device with which one need be concerned. With an indexed sequential file, however, there are three distinct areas: the prime area, index area, and the independent overflow area.

(1) Prime Area. As in a sequential file and a direct file, that area designated as the prime area will contain more than the data or file. It is divided into three areas: index area, overflow area, and the data area. The track index and overflow areas reduce the number of tracks which will be available for storing data on the file.

(2) Index Area. In addition to the track index, there will be at least one other index. Like the track index, they are created by the system at the time the file is created or reorganized. For each track within the prime area two entries will be made in the index. The first entry will be the home address of the data track and the key of the highest record on the track. The second entry is used when overflow occurs and will contain information pointing to the location of the overflow record.

(3) Overflow Area. The overflow area is optional; but if used, an area must be reserved for it. The size and actual location is determined by the programmer. The only restriction is that the overflow area must be on the same type of DASD as the prime area. This is not to say that it must be on the same device as the file, but it must be on the same type of device.

(4) File Maintenance. Records can be added to and deleted from the file without creating a new record. If the track where a record is to be stored is full, the system will shift information to the overflow area. On the other hand, if the track is not filled, the new record will be stored in the proper sequence determined by the key. When overflow occurs, the system will establish the necessary changes to the track index. The programmer will not handle the file any differently when you are accessing the file and overflow has occurred. The only difference will be in access time, wince it takes longer to perform a search after overflow has occurred and the desired record has been located in the overflow area. Deleted records will remain on the file and will be flagged to show that the record has been deleted. The deleted records and those records on overflow will remain on the file until the file is reorganized. At that time, the deleted records will be removed from the file, the records in the overflow area will be moved to their proper locations in the prime area, and new indices will be constructed. Periodically, files will require reorganization for three main reasons:

(a) The overflow areas are full.

(b) Additions increase overall run time when searching the file randomly.

(c) The prime area contains too many deleted records.

The operating system will keep the necessary statistics for reorganization. These statistics will be maintained on the DASD where they will be available to the programmer at any time. Information accumulated is:

Number of overflow areas that are full.

Number of unused tracks in the overflow area.

Number of reference to non-first overflow records.

Number of records marked for deletion.

These statistics, along with the information kept on deletions, will be useful in deciding when to reorganize.

(5) File Update. File update can be performed at the same time as file maintenance. There is no need to separate the functions. Since the updated record will be returned to its original location, no backup will be created. Necessary procedures must be designed into the system to accomplish recovery.

(6) Use of Indexed Sequential Files. This file structure tends to be most widely used since it provides the most versatile accessability to the information in the file. It is used primarily in three applications which have a mixed retrieval requirement, some sequential or tape-like retrieval, and some random retrieval. Looking at a payroll application as an example, at the end of the month there is a requirement to prepare a check for each individual on the file. Each record on the file would have to be retrieved in order to prepare a check for each employee. An additional requirement which must be supported requires the file to be updated at the time transactions occur. On-line terminals will be posted in the accounting department to post changes as they occur. This requirement must be supported by a random file. Now we have a mix; indexed sequential file organization would allow transactions to be posted randomly, but will also permit sequential processing at month end to produce the checks.

(7) Programming Aspects. Three areas must be reserved on the direct access storage device for this organization, where the other two types need only one. There must be an area for the data (prime area), an area for the indices (index area), and an area for overflow (overflow area). The entire prime area will not be available for storing data, but will be reduced by the space required for indexing. Since the system expects to find indices at the beginning of each track, the prime area must be aligned on whole track boundaries.

Section 4. FLOWCHARTING THE NEW SYSTEM

3.21. INTRODUCTION. One term that is frequently used in the Army is "The Big Picture". This term has wide application in the military, for all of us must be concerned with the big picture of the job and mission of our modern defense systems. This big picture concept is a necessity in order to see the interconnecting relationships in any large problem. By now, I feel you will agree that the design of any automated data processing system which utilizes a computer for its central component fails within the definition of a rather large problem. It is well to remember that a data processing activity may be costing half of a commander's budget. From this viewpoint, you can see why a well-designed, efficient system is necessary. Flowcharting the new system is concerned with the development of specifications which will achieve all the requirements set forth by management. A flowchart of the new system is a very important tool available to the systems analyst, if it is developed correctly. It allows the analyst to experiement with different approaches and methods to solve data processing problems.

3.22. DEFINITION. Probably the best explanation of a flowchart of the new system is -

> A FLOWCHART OF THE NEW SYSTEM IS A GENERAL PICTORIAL REPRESENTATION OF THE WAY IN WHICH YOU PROPOSE TO DESIGN A NEW SYSTEM.

The purpose of the flowchart of the new system is to show the recommendations for change to insure that the system will function as it is supposed to. The flowchart of the old system dealt with reality, the way the system is actually functioning; and it showed in great detail all the actions taking place. Now, you are preparing to solve the problems found in the old system, and, at the same time, implement the new requirements set forth by management. Basically the flowchart of the new system will show the source input, final output, the files used, and associated computer runs needed to solve those requirements. Since the flowchart shows the flow of data through all parts of the system and the relationship between them, the emphasis is on the media involved and the work stations through which they pass.

3.23. REFERENCE MATERIAL. Perhaps without realizing it, you have already started building the new system. The effort really started with the study of the present system. The main interest at that point was to determine the good parts of the old system and the areas where problems existed. With the beginning of the design effort for the new system, a statement of the functions and associated tasks which must be satisfied was developed. Finally, a listing of the data elements needed in the new system was begun. The basic documents developed by the design team which provide this information and were used as reference documents during the preparation of the flowchart of the new system are listed below:

a. The flowchart of the present system.

b. Statement of the new requirements.

c. Statement of each function, associated task, and action, together with the code "3" document associated with each task and action.

d. I/O DAS for the new system.

These documents play an important part in design effort. For example, a study of the flowchart of the present system is the starting point of the design, based on the following formula:

$$\text{NEW SYSTEM} = \left[\begin{array}{l}\text{OLD SYSTEM FUNCTIONS,}\\ \text{TASKS, ACTIONS, AND/OR}\\ \text{DOCUMENTS WHICH ARE}\\ \text{STILL GOOD.}\end{array}\right] + \left[\begin{array}{l}\text{REVISED PROBLEM}\\ \text{AREAS FROM OLD}\\ \text{SYSTEM.}\end{array}\right] + \left[\begin{array}{l}\text{NEW FUNCTIONS,}\\ \text{TASKS, ACTIONS,}\\ \text{AND DOCUMENTS.}\end{array}\right]$$

The first two elements of the formula can only be determined by analyzing the flowchart of the present system.

3.24. FACTORS TO BE CONSIDERED DURING FLOWCHART PREPARATION. There are certain factors that an analyst must consider during this phase. Each one will have a definite impact on the final product or solution to the problem and on whether or not the results will be productive.

a. Equipment Limitations. This factor must be considered in one of two ways, equipment is either available or it is not. The situation may be that new hardware must be acquired, so the proposal will include hardware requirements. On the other hand, the new system may be a command-unique system and must be designed to fit an existing hardware configuration. There will also be some cases where the hardware presently installed is not the type needed to satisfy the requirements; e.g., a tape system is available when the new requirements specify a direct access storage capability.

(1) Existing Hardware Configuration. This situation can be the most constraining because the system must be designed to "fit" the present environment. Physical characteristics of the hardware will impose constraints. A certain amount of memory will be available for the application programs, and this will have a bearing on how much can be accomplished within a given run. Another constraint on the design effort will be the types of I/O devices available - this will have a direct influence on file structure and organization. If the only I/O devices available are tape and card, it is impossible to support an indexed sequential file structure, because that

structure requires a direct access storage device. The last constraint has to do with time. The question of availability of time on the hardware must be addressed. The equipment may have already reached the saturation point and there is simply not enough time on the present equipment to run the programs. The new requirements may necesitate an upgrade, or an existing system may be deleted. In some cases, there may be a strict time requirement which may impact on such things as the selection of I/O devices, file organization, blocking files, etc.

(2) No Hardware. If the installation is starting without any equipment, consideration must still be given to equipment limitations. The proposal will contain recommendations as to the type and quantity of hardware necessary to support the system. When developing the recommendation, the design effort must be realistic. If the system requires direct access storage devices, the system should be designed with that requirement supported. If, however, the system can be supported by tape devices, the system should take this into consideration. Examine time constraints and design the system accordingly. If the situation calls for a 12-hour cycle, a proposal should not be developed which would accomplish everything in one run with a total run time of 15 minutes. How would the equipment be used during the rest of the time. The commander will be paying for the equipment even though it is not being used. Finally, the requirements may be such that no equipment exists or is readily available which will satisfy them. This situation requires additional research and development and would definitely extend the installation time. Examine these types of requirements to see if there are alternatives to solving the problems or supporting the requirement.

b. Priority of Processing. This factor deals primarily with sequencing. The sequence of the action to be accomplished and the sequence of the transactions to be processed in the system have an impact on the design. The purpose of the system is to furnish data, and that data should be as current and as accurate as possible. this is priority of processing which has an effect on the total system design from design of master files to design of final output.

(1) Master File. The first area to be examined is the relationship between the master files and the transactions in the system that will be used for file maintenance and updates. Various transactions which affect the content of the master files will exist in the system. The question of what bearing the sequence of processing these different transactions will have on the master files must be considered. Processing multiple types of transactions against the same file can lead to false impressions, or can allow false results to occur. Using the supply system as an example, each line item on the master file has a balance on hand. The item information, including the balance on hand is naintained on this file. At least three types of transactions affect the master file, primarily in the area of the balance on hand.

(a) Requisition. Submitted by an organization requesting supplies. Reduces the balance on hand.

(b) Receipt. Submitted by a supply depot reporting receipts or additions. Increases the balance on hand.

(c) Inventory Adjustment. Submitted by a supply depot to report unusual losses or gains. Can either reduce or increase the balance on hand.

The question is, what sequence applies? It would appear that the balance on hand should reflect the most current information before the requisition is processed. Thus, the requisition should be processed last. Now look at the receipt and the inventory adjustment, does any specific sequence apply to the processing of these transactions? If an inventory adjustment were processed before a receipt against the same item, it could produce a false negative balance. In this regard, management must clarify when the system is to recognize a negative balance as an error. If management states that the sequence of processing will have to be - receipts, inventory adjustments, requisitions. If management states that the system is to consider negative balances as errors only after all receipts and inventory adjustments have been processed, there is no impact if the inventory adjustments are processed before the receipts.

(2) Outputs. By first identifying each output associated with each step or run, a determination can be made as to what input is required and if it is available. Availability of input has a profound effect on the sequence of the processing steps. The condition of the inputs will also affect sequence. Before a given output can be produced, the input may need additional processing if it is not acceptable; for example, source input may need to be keypunched because the documents are not in a machine-readable format. The input would have to pass through a keypunching step before the output could be produced, resulting in a sequencing of steps. Currency and validity of the input information must also be addressed because the accuracy of the input determines the accuracy of outputs. At times, an output is the result of data produced in more than one processing step. In this case, provisions must be made to hold or store the data temporarily until the later processing steps have been completed. Thus, intermediate files are needed within the system. An intermediate file is a means of transferring information from one step to another step within the same run. An example of an intermediate file would be the transaction file created when transferring data from cards to tape as shown in Figure 3-15.

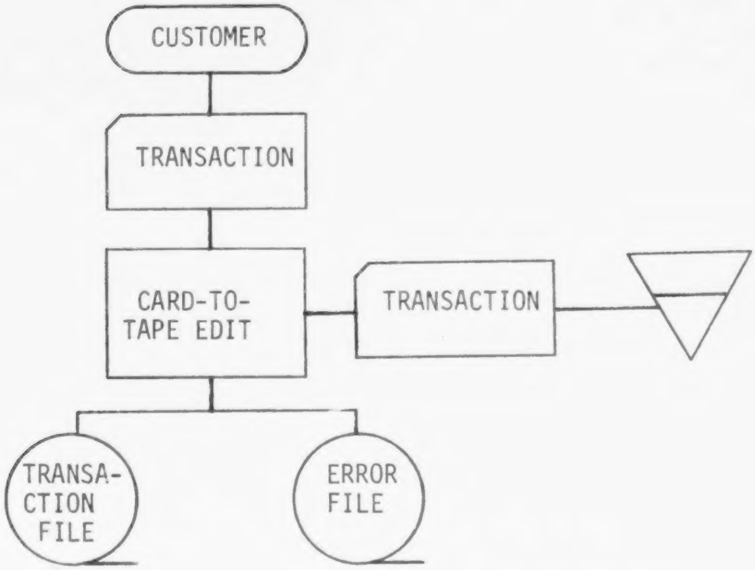


Figure 3-15.

The transaction file being created on tape is an intermediate file. It is created as a result of the edit step, changing the media from card to tape to take advantage of the speed of tape. Once the cycle is finished, the need for the file no longer exists. A new transaction file is created for the next cycle.

c. Retrieval Requirements. The last factor deals with how the customer will use the system. The main point here is how the files are going to be used. Both input and output requirements must be considered. Examine the input to determine how it is received and the output to determine how it is to be dispatched. Basically there are three methods; batch mode, on-line mode, or a combination of batch and on-line modes.

(1)Batch Mode (Figure 3-16). In this mode of operation, the transactions are accumulated and processed together, in a batch. Several transactions affecting the same master record will be handled or passed through the system at the same time. This type of operation calls for sequential file organization. An example of batch mode is the producing of payroll checks for each individual on the master file, as shown below.

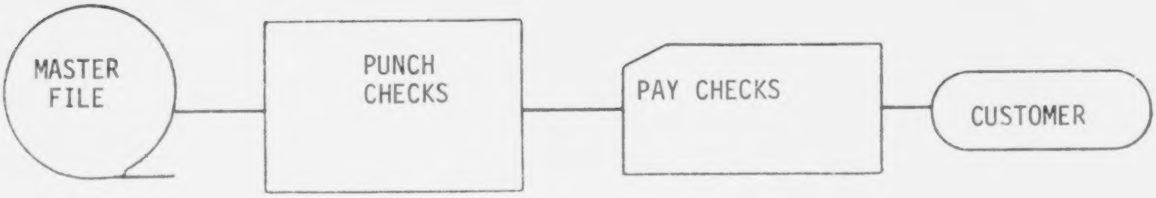


Figure 3-16.

(2) On-line Mode (Figure 3- 17). In the on-line mode, each transaction is processed as it is received. The operation calls for a direct file organization. The system must handle each transaction as it arraves; retrieve the applicable master file record; and take the necessary actions. An example of this would be a point-of-sale system. As a sale occurs, the information relating to the item is updated to reflect the change in available stock. The cash register used is an on-line terminal under the control of a host computer.

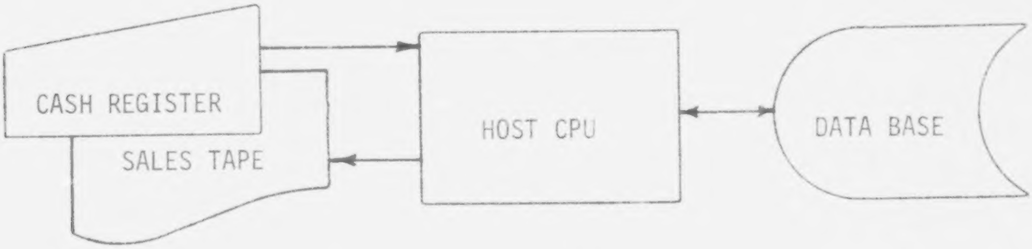


Figure 3-17.

(3) Combination Mode. In this mode, both batch and on-line requirements exist. Transactions may be accumulated and pass through the system at the same time (batch mode) while the system must still have the capability of responding to inquiries initiated at any time. Using the payroll system as an example, at the end of the month there is a requirement to produce paychecks for each individual. This requires a file which can be processed sequentially. There is also a requirement to support a terminal within the accounting office to handle inquiries against individual records throughout the month. The inquiries will be made against the same file from which the pay checks are created. Processing inquiries calls for a randomly accessible file. In order to satisfy both requirements, the system needs an indexed sequential file, one which can be processed both sequentially (for pay checks) and randomly (for inquiries).

3.25. SYMBOLS. Only those symbols and definitions which conform to the American National Standards Institute (ANSI) will be used in preparing the flowchart. The symbols and their definitions are contained in paragraphs 2.40 and 2.42 of this text. With the exception of the decision symbol, the same symbols used in flowcharting the present system are used in flowcharting the new system. A flowchart of the new system does not have the detail of the present system flowchart. The definition of the flowchart of the new system states that is is a general pictorial representation. Because the decision symbol would show too much detail, any decisions are considered as part of the processing taking place.

3.26. CONSTRUCTION GUIDELINES. The flowchart of the new system is based upon the good parts of the old system, revised old problem areas, and new requirements. In dealing with the new system, a proposal or recommendation is being developed. The systems analyst is now dealing with what should take place to make the new system work properly. The following steps apply to constructing a flowchart of the new system.

a. Identify the functions, associated tasks and actions, and supported final outputs (code 3 documents). The purpose of the new system is to satisfy the functions in support of the mission of the organization. Outputs have been designed and approved by management. Now the system must produce those outputs. Prepare a list in the following format which associates the outputs with the functions:

FUNCTIONS	TASK/ACTIONS	DOCUMENTS (CODE 3)

b. Identify the sequence in which the functions must be accomplished. Examine the relationship of the functions and identify those functions which depend upon other functions. Identify any time requirements that may have been established by management to satisfy a given function. Based upon these findings, establish the sequence in which the functions must be accomplished.

c. Identify the sequence in which the tasks and actions associated with each function must be accomplished. First, examine the relationship of the tasks and actions within a single function. Identify the tasks and actions within that function. Examine each function as a separate entity in the manner just described. Next, examine the relationship of the tasks and actions within that function. Examine each function as a separate entity in the manner just described. Next, examine the relationship of the tasks and actions in one function to the tasks and actions associated with other functions. Any relationship in this category may force a realignment of a function's sequence or a separation of the actions within a function. The first examination deals with the relationship of tasks and actions within functional lines; while the second deals with the relationship of tasks and actions across functional lines. Finally, identify any time requirements that may have been established by management to accomplish a given task or action, or to produce an end product. Based upon these findings, establish the sequence in which the tasks and actions must be accomplished.

d. Identify what is to be accomplished in a given run. Two things must be determined at this point. What is to be included in the program, and how much can be done within the program. Each processing block on a flowchart of the new system will be come an applications program. The "what" is identified by examining the list of functions and associated tasks and actions. How much that can be accomplished in a single program will depend on equipment limitations.

(1) Amount of available memory.

(2) Types and numbers of I/O devices available.

Based on the established sequence, identify the basic processing steps in general terms. Determine the amount of memory needed to support each of the basic processing steps. It may be necessary to split up a task or action into sub-tasks or sub-actions simply because there is not enough memory available to accommodate the resulting application program. At this point then, we can consider an example of the associated computer runs within the flowchart (Figure 3-18).

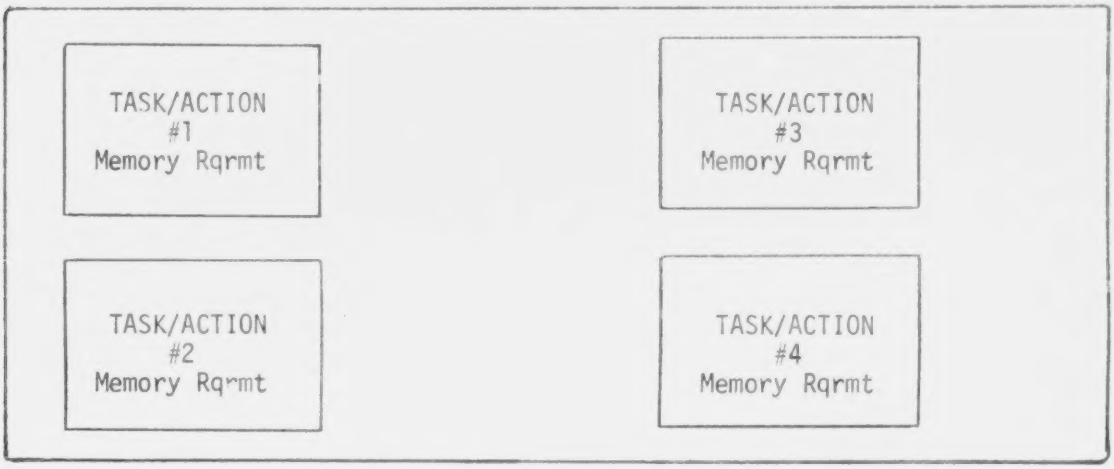

TASK/ACTION #1 Memory Rqrmt
TASK/ACTION #3 Memory Rqrmt
TASK/ACTION #2 Memory Rqrmt
TASK/ACTION #4 Memory Rqrmt

Figure 3-18.

e. Identify the final outputs (code 3) associated with each run. There will be outputs associated with each processing step. What outputs will depend upon what task or action is being accomplished in the processing step. The listing of functions, tasks and outputs (paragraph 3.26a) will provide the relationship. Each output will have to be supported by an I/O device. The type of device will depend upon the media; i.e., tape, DASD, hard copy, etc. For each processing block, complete the outputs as shown in Figure 3-22. From this illustration it can be seen that a printer, card punch, and magnetic tape unit are needed. The number of devices required and available, by type, must be known. For instance, there may not be a tape unit that is available, but a DASD is. You would then need to determine the effect a disk file would have on the system. Also, it is apparent that the processing step produces two reports. If only one printer is available, how will the second report be accommodated? Will it occur at the same time as the first printing? Again, it may become necessary to split up a task or action into smaller sub-tasks or sub-actions based upon the non-availability of I/O devices.

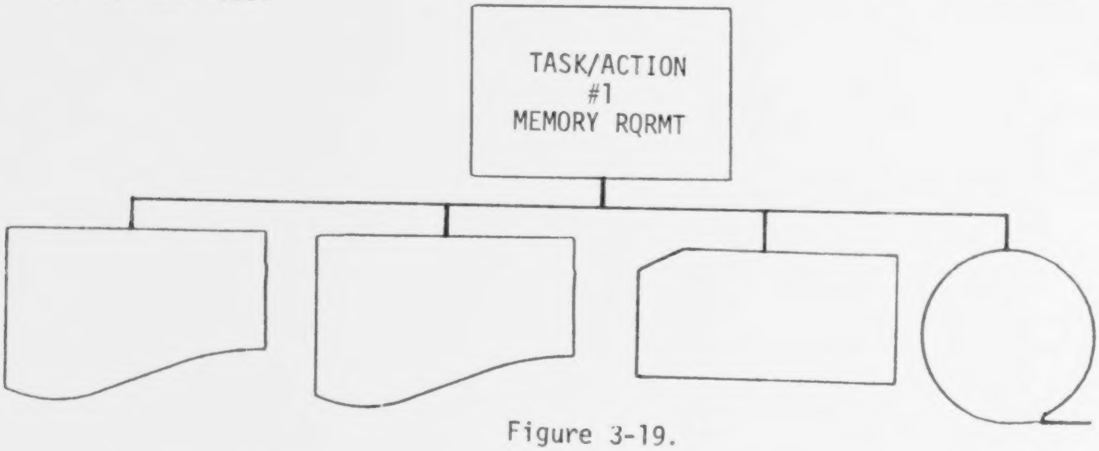


Figure 3-19.

f. Identify the input data elements necessary to support the output for each run. The results of the preceding step provide the outputs associated with each processing block. The DAS for the new system will identify the data elements that must appear on each of the outputs. Now the analyst knows what data elements are necessary as inputs (Identifying, Quantitative, Reported, and Constant data types). For those output data elements coded as Generated, a further analysis must be made to learn what data elements are required to generate the item. These data elements will be your input.

g. Determine what input documents (code 0) will be available to support the output. The input data elements have now been identified. Now the question is, where are they going to come from? The I/O DAS will again be the reference document. It provides a data element-to-data element relationship, or shows on which document the data will be available. Additional facts as to the required media and sequence must be considered. Media will have a bearing on the type of I/O device required. Also, the media and sequence of source documents may require some preceding processing steps. Consider a processing step where a master file is being updated; the output of the run includes a shipping document which is sequentially organized. This would require the input to the run to be in the same sequence as the output; thus, a SORT step must precede the run. Figure 3-20 illustrates this case.

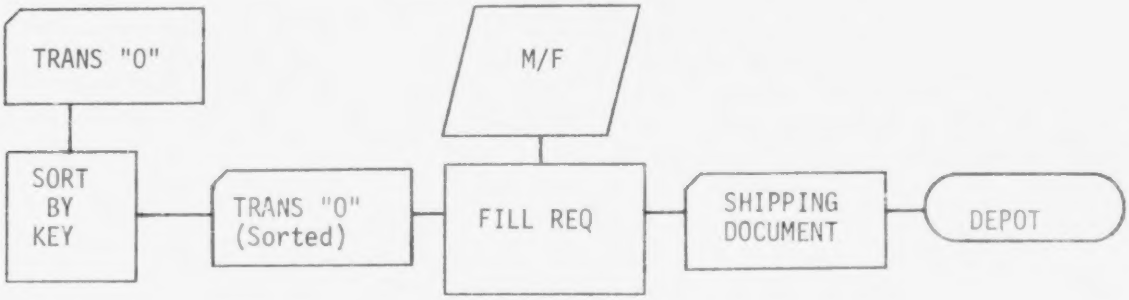


Figure 3-20.

h. Identify those runs which require a file and determine the contents of the file. Some processing steps will require input information that is not contained on a source input document. For example, in Figure 3-23, additional information is needed to show how much of the item is on hand. This extablishes the need for a file within the system. In this case the file will be a basic record (code 1) because the information is needed for day-to-day operations. In addition, the data elements to be retained within the system on master files must be determined. Following are some basic considerations when determining contents.

(1) Fields coded as Identifying and Quantitative on the I/O DAS are primary candidates for inclusion. All or almost all of these fixed elements will be retained on master files to meet application-oriented output needs.

(2) A small number of items coded as Reported and Generated on the I/O DAS may be retained on the master files to meet current and future information storage and retrieval needs.

(3) Future output needs should also be considered to avoid the early obsolescence of the system's master files.

(4) Redundancy of data among two or master files is to be avoided, if at all possible. This will minimize the size of the system's files and will reduce repetitive posting or updating of the same data on different files within the system.

i. Selection of data elements. The basic document to assist in the selection of data elements to be included in a file will be the I/O DAS. The output data elements necessary for the processing step and the data elements which will be available through source input documents have already been determined. The remaining data elements needed by a processing will be found in a file, or files; either permanent (basic record) or temporary (intermediate record). It is during this step that the need for intermediate file (code 2) becomes apparent. The purpose of an intermediate file is to carry data from one processing step to another within the same cycle. The following conditions may result from intermediate files.

(1) Equipment limitations. There may not be enough memory available to accomplish a complete action in one processing step. An intermediate file is needed to carry information from the processing step where an action is begun to the processing step where the action is completed.

(2) Sequence of Processing. Information generated in one processing step is needed in another step which will be run at some later time.

(3) Programming considerations. The information extracted from a master file will be used as an input to a future processing step eliminating the need to pass the entire master file. Changing from a slow input medium to a faster medium should be considered.

Figure 3-21 illustrates a need for intermediate files. In the example, there are two intermediate files; requisitions on punched cards, and the purchase file on magnetic tape. The medium had to be changed on the requisition because the original document was not in a machine readable format. The purchase file is needed to transfer information about purchase orders from one processing step to another processing step. Once the purchase order is produced, there will be no further need for the purchase file.

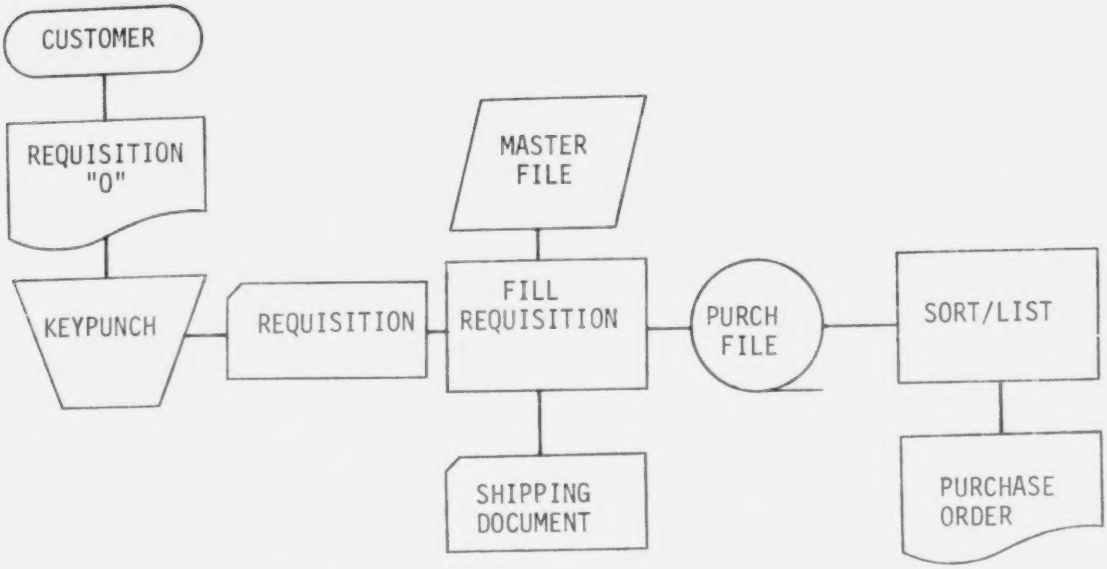

CUSTOMER
REQUISITION "O"
MASTER FILE
KEYPUNCH
REQUISITION
FILL REQUISITION
PURCH FILE
SORT/LIST
SHIPPING DOCUMENT
PURCHASE ORDER

Figure 3-21.

j. Select file structure and appropriate supporting storage media. Selection of file structure deals with determining the method of file organization to be employed. The requirement that has the greatest impact on file organization is the retrieval requirement against the file. For example, a sequential file organization would be a poor selection for an on-line batch system. The selected file structure also has a definite impact on the choice of media. A sequential file can be stored on disk or tape, so there is no real problem where media is concerned. Random or indexed sequential files both require a direct access storage device, however. Consideration should also be given to consolidation of files and the effect this will have on processing. Compare the effect of a single file supporting the system, instead of several files supporting the same system. For each case develop an estimate of the following items:

(1) Record volume.

(2) Activity of each file, type of record, and member field.

FILE SPECIFICATION

For use of this form, see TB 18-111; the proponent agency is DCSOPS.

1. DATE YYMMDD

2. ID EPRW001	**3. TITLE** Employee Payroll System Master File		
4. FILE DESCRIPTION Contains permanent pay and employment data used in preparation of weekly payrolls.			
5. STORAGE MEDIA Disk	**6. RECORD FORMAT** Fixed	**7. FILE ORGANIZATION** Sequential	**8. SECLAS/PRIV** U/Y
9. TAPE REC MODE N/A	**10. LABEL FORMAT** Standard	**11. SEQUENCE** SSN	**12. PARITY** N/A
13. TAPE BPI/TRACK N/A	**14. MIN-MAX SPACE REQ** 2000/2500 records	**15. BLOCK FACTOR** 10	

16. RETENTION

a. UPON TERMINATION WHEN DET NLN FOR MSN RQMTS	**b. UPON UPDATE** 3 cycles	**c. DUPLICATE(S)** 5 years

17. RECORD TITLE	18. RECORD ID	19. % FILE
Employee Master Record	EMR001	100

20. REMARKS

None

Figure 3-22

DA FORM 4737-R, APR 83 REPLACES DA FORM 4737, NOV 78, WHICH IS OBSOLETE.

RECORD SPECIFICATION For use of this form, see TB 18-111; the proponent agency is DCSOPS.	**1. DATE** YYMMDD

2. ID EMR001	**3. TITLE** Employee Master Record

4. DESCRIPTION Contains all necessary payroll and employment data about individual employees.	**5. LENGTH** 124 **6. SECLAS/PRIV** U/Y

7. POSITION	8. FIELD	9. FIELD TITLES	10. CLASS	11. LENGTH	12. REMARKS
1-9	1	SSN	N	9	KEY
10-29	2	Employee Name	AN	20	
30-45	3	Unit of Assignment	AN	16	
46	4	Pay Code	A	1	
47-48	5	Years Service	N	2	
49	6	Special Pay	N	1	
50-51	7	# Dependents for Taxes	N	2	
52	8	Insurance (Life)	A	1	
53-54	9	State of Residency	A	2	
	10	Allotment			
55-74	10.1	Name	AN	20	
75-80	10.2	Amount	N	6	
81-124	10.3	Address	AN	44	

Figure 3-23

DA FORM 4738-R, APR 83 REPLACES DA FORM 4738, NOV 78, WHICH IS OBSOLETE.

(3) Processing method. This method of examination will provide a means of selecting the best file structure. In doing so, try to minimize processing time; minimize the total size (number of characters) held within the system or master file; and maximize the simplicity of design in the system.

Following through with the actions involved at this step in development will enhance the design of the new system and the impact the files will have on the system as a whole. Remember, these actions include studying the retrieval needs or types of processing required for the application; investigation of equipment costs or other constraints imposed by management or the environment; developing as many possible master file structures and media solutions as time will permit; and conducting trade-offs and selecting the best alternatives. Be prepared to discuss possible solutions with management.

k. Format records internally and document the results. Associated with this step are four basic actions:

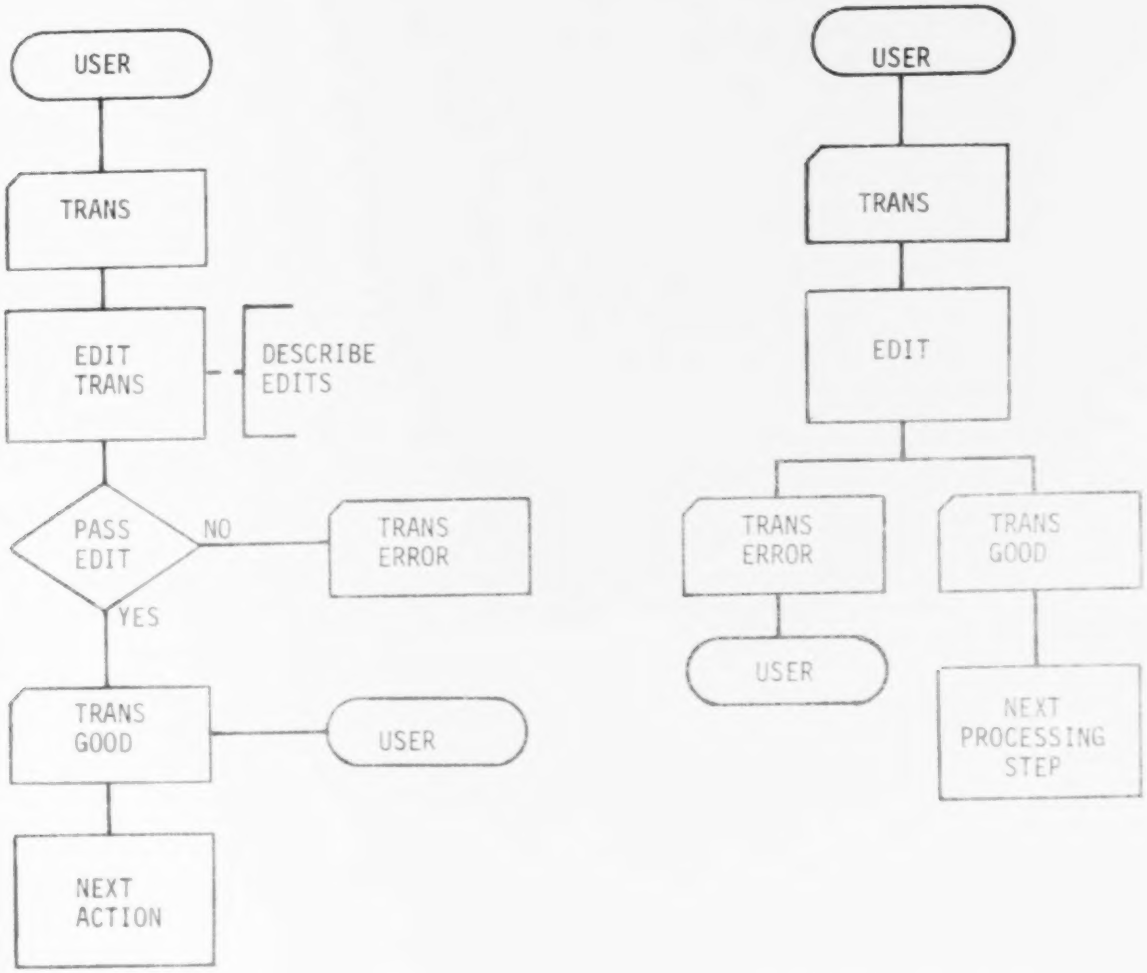
(1) Lay out the logical records for each file in the system by using DA Form 4737 (Figure 3-22) and DA Form 4738 (Figure 3-23). The DA Form 4737 is used to define the file characteristics, while DA Form 4738 is used to define what a record looks like for card, tape, or DASD files. The following notes should be considered.

(a) Indicate all data which describes the record (record ID, title, description, length, etc.) in the appropriate blocks of DA Form 4738. Information describing the file (file ID, title, storage media, file organization, record format, etc.) will be shown on DA Form 4737.

(b) If no field in the record is noted as the file key, the first field in the logical record will be assumed to be the key. In the sample in Figure 3-23 "SSN" is the key. This fact is noted in the remarks column of the form.

(c) Fields which are in packed format are to be so noted in the Remarks column.

l. Prepare the flowchart of the new system. The final step is to prepare the pictorial representation of how you propose to solve the problem. A flowchart of the new system does not contain the detail found in the flowchart of the present system. It deals with the general, not with the specifics of the system; therefore, it is unlikely that a decision symbol will ever be found in a flowchart of a new system. The new system chart includes primarily the main processing steps and the associated inputs and outputs. The example in Figure 3-24 illustrates the differences between a present system flowchart and a flowchart of a new system.


USER
TRANS
EDIT TRANS
DESCRIBE EDITS
PASS EDIT
NO
YES
TRANS ERROR
TRANS GOOD
USER
NEXT ACTION
USER
TRANS
EDIT
TRANS ERROR
TRANS GOOD
USER
NEXT PROCESSING STEP

FLOWCHART OF PRESENT SYSTEM

FLOWCHART OF NEW SYSTEM

Figure 3-24.

Section 5. SYSTEMS RELIABILITY

3.27. INTRODUCTION. a. Internal controls in an ADP system vary widely, depending on the type of application and its complexity. The actual control techniques to be employed by an organization depend upon individual conditions and requirements, and the knowledge of the personnel responsible for the design and implementation of the system.

b. The changes in methods of preparing financial records do not relieve the financial manager of the responsibility for internal control. On the contrary, increased emphasis should be placed on internal control because of the centralization and concentration of data processing functions within a separate organization. Accuracy of processing is ultimately the user's responsibility - not data processing's. Consequently, the financial manager should be aware of the available controls in an ADP environment.

c. We should also look at ADP controls from the viewpoint of the auditor. The American Institute of Certified Public Accountants (A.I.C.P.A.) has designated field standards which clearly fix any auditor's responsibility in determining the adequacy of internal control. These standards are generally accepted by all auditors. Two field standards which are most significant to internal control considerations are:

(1) There is to be a proper study and evaluation of the existing controls as a basis for reliance thereon and for the determination of the resultant extent of the tests to which auditing procedures are to be restricted.

(2) Sufficient competent evidential matter is to be obtained through inspection, observation, inquiries, and confirmations to afford a reasonable basis for an opinion regarding the financial statements under examination.

d. Before we look at specific ADP controls, we should examine the definition of internal control and the basic difference between a manual and an automated environment. Internal control is "a plan of organization and all coordinated methods adopted within a command to safeguard its assets, assure accuracy and reliability of its records and reports, promote operational efficiency, and encourage adherence to prescribed managerial regulations and directives." Although these objectives do not change in an automated system, new control techniques are needed for their accomplishment. In a manual system, the two most important elements of control have been (1) appropriate separation of duties; and (2) a well-defined system of authorization. This means an almost total reliance on human alertness and care in order to provide for a well-controlled system. However, in a computerized system, there is a reduction in manual processing and an apparent loss in the areas of human judgement and separation of duties. We shall see that this apparent loss can be more than made up through proper utilization of today's more sophisticated hardware and corresponding use of ADP controls.

3.28. THE AREAS OF INTERNAL CONTROL. To aid in understanding how a balanced control system is established, we will divide ADP controls into seven logical areas. The environment that must be controlled and the areas of control are as shown in Figure 3-25.

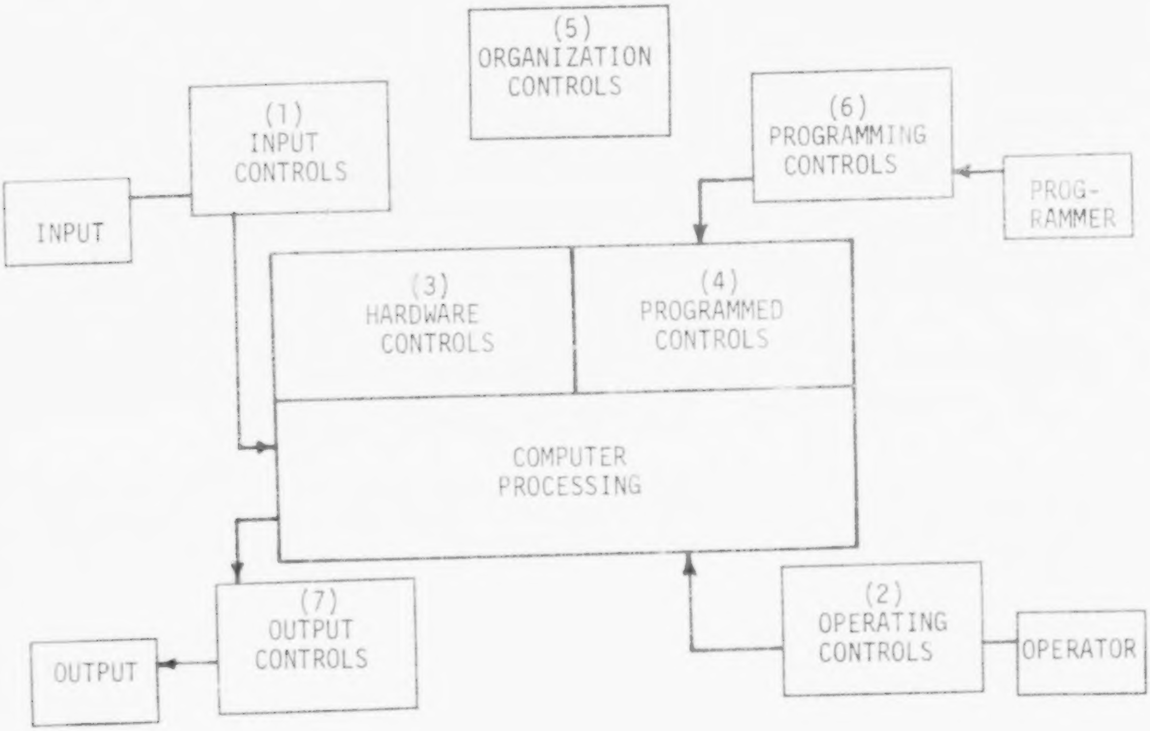

(5) ORGANIZATION CONTROLS
(1) INPUT CONTROLS
INPUT
(6) PROGRAMMING CONTROLS
PROG-RAMMER
(3) HARDWARE CONTROLS
(4) PROGRAMMED CONTROLS
COMPUTER PROCESSING
(7) OUTPUT CONTROLS
OUTPUT
(2) OPERATING CONTROLS
OPERATOR

Figure 3-25.

3.29. INPUT CONTROLS. Looking at the input phase of a common computer operation, the areas of possible error are as shown in FIGURE 3-26.

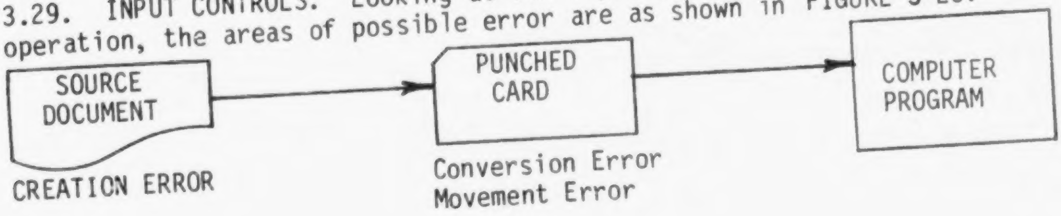


Figure 3-26.

Input controls are intended to preclude the possibility of the types of errors shown in Figure 3-26. Therefore, the objective of input controls is to assure accuracy and completeness of input. The responsibility for these controls is shared by the user and data processing personnel. The following are control techniques that may be used to control each area of possible error.

a. Creation of Input. The controls here must insure accurate manual processing. In addition to the control techniques commonly used in manual systems, several control techniques that take on additional importance in an automated environment are discussed below.

(1) Simplified and Standardized Forms. Well-designed forms impose procedural controls on the creation of data. For example, a part number to be written on a document may have to be written in a special position which contains the exact number of spaces needed for that part number. The spaces can be marked in such a way that a part number containing fewer or more characters than the required number of characters will be detected by the person who writes the part number.

(2) Precoded Source Documents. To reduce the possibility of coding errors, some documents may be precoded with frequently used codes. For example, inventory source documents could have stock numbers for the most frequently used items of stock already coded on the form. Then, when filling out the form, the writer needs only to circle the appropriate stock number and write in any variable data required. Another example would be a precoded vendor number, or customer code.

(3) Visual Review. Review procedures require examination of input data before it is converted to machine-readable form. The type of data that would be appropriately checked would include identification or descriptive information such as controlling codes, customer names, price extensions and footings. If the source document is extremely complex, it is often desirable to manually transfer the appropriate data to a simplified or standardized form.

b. Movement of Input. Control procedures in this area must insure against the loss or duplication of data between its point of oriogin and the first computer run. System controls deal with the movement of data between the first computer run and the ultimate disposition of the output.

(1) Anticipation Control. If we can anticipate what movement data must follow by using document numbers or data compatibility tests, we can control movement by taking note of exceptions to what is anticipated. This types of controls can be readily implemented in the first computer run as programmed controls, but they are covered here in their logical progression.

(a) Sequence Check. Serial numbers are printed or stamped on each document. This technique is particularly adaptable to documents such as checks, vouchers, or other sensitive documents which must be individually accounted for. Input documents so identified may be checked either mechanically or electronically to insure proper sequence and/or to reveal any missing documents.

(b) Document Register. A variation of sequence checking can be a manual or computerized record of all possible document numbers. As data is processed, the document numbers are "cancelled", and a review of the record would disclose any missing document numbers.

(c) Matching. Reassembling duplicate documents at a predetermined point to insure completeness of processing is the basis of this control technique. When all copies of a document have been reassembled and matched at the control station, the presence of all copies indicates completed processing. The technique of matching is particularly useful when multiple copies of a document are used to perform specific functions at different locations, such as in a receiving department to show arrival of goods, and in a purchasing department to indicate a bill which has to be paid. When all of the copies are reassembled later, this is indicative that the goods have veen paid for, and that the organization has physically received them. If the source documents have bbeen converted to another medium such as punched cards; the reassembly, matching, and checking can be done automatically.

(2) Batch Controls. Batching is the process of accumulating documents into an economic processing unit. Batching can control the movement of input when totals of the batched data are used. These totals are accumulated prior to conversion. Totals may be hash totals, which are sums of meaningless data (account number, etc.); or control totals, which are totals of meaningful data (dollar amounts, quantity, etc.). After data conversion, the batch totals are recomputed using either EAM or a computer and this total is compared to the original total for the batch. If the totals do not

agree, input documents or media may have been lost subsequent to the original control total's computation and prior to the independent computation. The first step in reconciling the total's would be to recompute the original control total's computation and prior to the independent computation. The first step in reconciling the totals would be to recompute the original control total. If the discrepancy still exists, a detailed check of all source data must be made to locate the missing data. Batch totals can also be used to control the accuracy of the data conversion process, however, this will be explained later. The following rules should be considered in batching.

(a) Individual batches should be numbered to facilitate reconciliation procedures. This is especially important when the integrity of individual batches is destroyed during processing.

(b) The number of items within a batch should be limited to a size which will permit ease of reconciliation and ease and efficiency in handling.

(c) Types of transactions included in any one batch should be as homogeneous as possible to provide consistent batch totals.

(d) More than one batch total should be used for each batch to clearly distinguish lost data from conversion errors.

(e) Where punched cards are used as input media it may be desirable to check batch totals in the initial step of an edit program. This procedure is intended to clean up data before it is used in other processing where high error rates cannot be tolerated. If this procedure is done in a manner which accounts for every transaction, it will account for the completeness of data through editing, but not through any further processing.

(f) To control completeness of data past editing, programmed verification of batch totals should be made by the computer.

(3) Cancellation Stamp. Occasionally the errors in movement of data are not lost data, but are duplicated data. To preclude the duplication of input data, a cancellation stamp should be used on all source documents at the point of data conversion. Once a batch of documents has been keypunched, it should be cancelled so that it will not be punched again. If a time stamp is used, it not only affords cancellation, but it also provides work or production control because the elapsed time in handling is identified.

c. Data Conversion. To prepare data for entry into the data processing system, the information on the original source documents is normally converted to a machine-readable form. This is done thorugh some sort of rekeying process which places the original data on a computer-acceptable input medium. As a result of this duplication of input creation, an additional step has been taken which introduces a possibility of errors occurring. We must control this process, to reduce the probability of error. The objective of data conversion controls, then, is to preclude or pinpoint erroneous conversion.

(1) Key Verification. To key verify punched cards, the cards are placed in a machine similar to a keypunch machine, and the data is re-keyed. If the data on the cards are different from what is keyed, a red light comes on to indicate a possible error, and the cards are notched on the edge above the column where the error occurred. The cards can then be repunched to correct the error. Experience has shown that key verification will not catch all errors. This is due in part to illegible source documents or to haphazard verification practices. The expense of key verification is rather large. For all practical purposes, 100% key verification of input data will double the cost of data conversion. So, the expense of key verification should be compared to the potential saving expected from the process. Cost of key verification can be reduced by verifying only the essential portions of the input information, rather than the entire transaction. Key verification may be eliminated in some cases in favor of strict edit procedures.

(2) Batch Controls. These controls can be used not only to control the movement of data, but also to control the data conversion process. This is accomplished by taking a batch total on critical numerical fields. If a batch total does not agree while other totals do, then there must be a conversion error in the field that is out of balance. This can be a very effective control technique since it accomplishes two objectives (control of movement and data conversion) at the same time. If an out-of-balance condition does occur, and after it is determined that it is not due to data loss, the batch must be examined in detail to locate exactly the transaction that was erroneously converted.

(3) Manual Edit. When a group of cards has been punched with some common data in certain fields, the cards can be visually verified by holding the deck up and looking through the appropriate punches. Also, if there is only a small amount of data, the cards can be visually verified if they have been interpreted.

(4) Prepunched Cards. Whenever input data contains certain fields that remain constant or have consistent variances, such as sequence numbers, the cards can be prepunched prior to data conversion. The variable data can then be punched into the appropriate fields. This technique precludes errors in the constant or consistent data fields.

(5) Turnaround Documents. As a variation of prepunched cards, turnaround documents are machine output that accompany a transaction to a point where it is returned with additional data. When the document is returned,

most (if not all) the information has already been punched. After any additional punching is done, the data is ready for machine input. This approach reduces the possibility of error by eliminating an additional data conversion process. An example of a turnaround document is a credit card bill where a punched card is returned with the remittance.

(6) Simultaneous Machine Input. As we have seen, the most effective way to increase accuracy in the area of data conversion is to reduce or eliminate the additional step of recreating source input on computer-readable media. In other words, to eliminate errors in the data conversion process, eliminate data conversion. This can be accomplished through the use of typewriters, bookkeeping machines, point-of-sale recorders, etc., which create both source documents and machine-readable media simultaneously. Therefore, with proper and complete controls (usually traditional) over the accuracy of the creation of source data, the input media must also be correct. Thus, we have eliminated conversion errors by eliminating conversion.

(7) Programmed Controls. Other controls to check the accuracy of input data can be programmed, usually into the first computer run. These controls include character and validity checks, self-checking numbers, and limit checks, and are discussed in detail under Programmed Controls.

3-30. OUTPUT CONTROLS. Considering the output phase of an automated operation the only types of errors possible would be the dissemination and use of erroneous output, or failing to correct and reprocess erroneous output. Therefore, the objective of output controls would be to insure the accuracy and completeness of output.

a. Determining Valid Output. The output should be tested for validity before it is used. This can be accomplished by the user and a control group if one exists. Control groups will be discussed under Organizational Controls.

(1) Visual Review. After the printed output is produced, the printouts can be visually reviewed by either the control group or the user to determine if the format is proper and if there is any "garbage' on the report.

(2) Balancing Control Totals. Control totals should be balanced to insure completeness of output in comparison to input. Control Totals should exist at least in the first computer run, and may exist in each successive run if a need for benchmarks exists.

(3) Detailed Test. As a final check of output, it may be desirable to take one or more transactions and trace their processing from their entry into the system to the effect they have on system outputs. This will test the accuracy of processing and may be done using fewer transactions as the programs and operations personnel prove themselves.

b. Suspensing Invalid Output. Once output has been determined to be inaccurate or incomplete, the output should be returned to the source of the error (the user for incomplete or erroneous data; ADP for program errors) for correction. However, to insure that the error is corrected, the data reentered, and the output returned to the user, a suspense should be established and followed up on if the corrections are not made. The responsibility for the suspense should be placed with the individual in each using activity who is responsible for originally reviewing the output for validity. This function can also be centered in a control group.

3-31. HARDWARE CONTROLS. Looking at the hardware phase of a computer system, the only sources of computer-oriented errors would be electrical or mechanical problems. An electrical circuit could short out, or a moving part could wear out. With today's advanced technology, hardware oriented errors are so uncommon that they are measured in errors per million or billion processes. It is very important that the hardware operate accurately, for if we could not rely on it, it would be very much worse than a staff of employees compounding errors at a very rapid speed. So the purpose of a hardware controls is to insure and monitor the electro-mechanical reliability of the computer.

a. Built-in Controls. These controls are built-in to the computer. Being built-in, they must be considered when originally acquiring the ADPE equipment. Once installed, the exercise of these controls would be through proper operating procedures when an error is indicated. Some of these control features are:

(1) Parity Checking. A parity bit is appended to an array of bits to make the sum of all the bits in the character set always even or always odd. For example:

	Bit	Even Parity		Odd Parity	
Check bit	C	1	0	0	1
Data bits	B	1	0	1	0
	A	1	0	1	0
	8	0	0	0	0
	4	0	1	0	1
	2	0	1	0	1
	1	1	0	1	0
Character		A	6	A	6

A parity check tests whether the number of "1" bits in a character array is either even or odd, depending upon the type of computer. If the number of bits is even and the computer requires odd parity, the machine will halt and a parity light will come on indicating that the machine encountered an electrical malfunction. This type of parity check is frequently called a vertical parity check and is found in almost all computers and most magnetic tape systems. The same type of check, but applied horizontally to a tape record is common in magnetic tape processing. The parity bits are added to the character or record when the character or record is created in core storage on magnetic tape.

(2) Echo Check. An echo check is used at points where data is transferred into or out of the computing system. Its purpose is to control the electromechanical functions of input and output. As this transfer is made a signal is sent back to the original source of the data. If there is any variation between the transferred data and the original data, a machine failure is indicated. This type of check takes on several forms.

(a) Read-write Check on Magnetic Tape. As information is recorded on magnetic tape, the record just written is immediately read and echoed back. This is compared against the record sent to be recorded to determine if the two are equal.

(b) Timing Check on Printer. A signal is sent to a printer to make print wheels advance to the desired printing position. The amount of actual advance is sensed, converted to the printing condition the position represents, and is compared with the original signal.

(c) Hole Count. The total number of holes read or punched in one column of a card at one station is compared with the total number of holes read from the same column of the card as it passes through another station. If the results of this check are valid, there is a high probability that the data has been transferred correctly.

(d) Disk Cyclic Check. This check is very similar to the tape read-write check. It is a procedure whereby data written to magnetic disk is compared, character by character, tothe data that should have been written from the computer's core storage.

(3) Character Validity Check. The bit configurations of characters read into main memory are tested to insure their acceptability as valid characters. This check insures that a valid character is being transferred, but not necessarily the correct character. An invalid character could pass a parity check, but not a validity check.

Bit	Valid	Invalid
C	0	0
B	0	0
A	0	1
8	0	1
4	1	1
2	1	1
1	1	1

(4) Reverse Arithmethc. A method by which a computer verifies a calculation. Under this method, a calculation of A times B equals T is checked by extending B times A equals X, subtracting X from T, and insuring that the result is zero.

(5) Dual Circuitry. Operations are performed twice on the computer, independently, and at the same time, and the results are compared. The space requirements and the cost of this duplication has nearly eliminated it from use, except in older, tube-type computers.

b. Preventive Maintenance. Another technique to insure the electromechanical reliability of the computer is to have a customer engineer from the computer manufacturer perform preventive maintenance on a regularly scheduled basis. This maintenance encompasses the prodding of the hardware looking for weak circuits or parts about to fail. This technique can prevent many serious problems before they happen.

3-32. PROGRAMMED CONTROLS. In a manual system, accuracy was dependent upon human alertness, care, acceptance of responsibilities, and division of duties. However, with the processing taking place within the computer, controls based on human judgement or separation of duties are no longer available. Programmed controls provide an alternative to these human controls. Programmed controls are controls are controls that are coded as part of the application programs in addition to the instructions which actually process the data. Programmed controls can be used to detect erroneous input data (input controls). The editing of input data should be done in the very first computer run of a system to minimize any later with accounting policies (accounting controls).

a. Character Check. This is an input control. It is used to check any one character or a number of characters in an input record. These characters can be checked for alphabetic versus numeric data, blank versus filled data, or any other data that should be consistent.

b. Validity Check. This is also an input control. If, for a given field, there is a limited number of codes, the input data is checked to see if it is one of the valid codes. Examples include transaction codes, month numbers, or even identification numbers, if a file of valid ID numbers is available for comparison.

c. Self-Checking Number. A system of adding a check digit to identification numbers (account numbers, stock numbers, etc.). The check digit is determined by performing some arithmetic operation on the number in such a way that the usual errors encountered in transcribing a number will be detected. An example of the "modulus 10" method of arriving at a self-checking number follows:

$$\begin{array}{lccccccccccc}
\text{Original number} = & \underline{6} & & 2 & & \underline{1} & & 3 & & \underline{5} & & \\
 & \text{x} & & & & & & & & 2 & & \\ \cline{2-12}
 & 1 & + & 2 & + & 2 & + & 1 & + & 0 & = & 6 \\
 & & & & & & & 2 & + & 3 & = & 5 \\
 & & & & & & & 20 & - & 11 & = & 9
\end{array}$$

Self-checking number 621539. Multiply the units position and every alternate digit by 2, adding the individual digits in the result and the digits not multiplied by 2, and determining the number necessary to raise the result to the next higher multiple of ten. Note that a check digit is not completely error proof. An alternate transposition would give the same check digit. However, it is unlikely that this will occur. The check digit does not guard against the possible assignment of an incorrect but valid code.

Another type of error check is "modulus 11". In this check, every digit is multiplied by a number starting with 2 for the units position and increasing by one for each additional position in the original number. The sum of these products is divided by 11 and any remainder is subtracted from 11 to give the check digit. An example follows:

$$\begin{array}{lccccccccccc}
\text{Original Number} & 6 & & 2 & & 1 & & 3 & & 5 & & \\
 & \underline{6} & & \underline{5} & & \underline{4} & & \underline{3} & & \underline{2} & & \\
 & 36 & + & 10 & + & 4 & + & 9 & + & 10 & = & 69 - 11 = 6\text{-}3/11
\end{array}$$

$$\begin{array}{lr}
\text{Modulus} & 11 \\
-\ \text{Remainder} & \underline{3} \\
\text{Check Digit} & \underline{8}
\end{array}$$

Self-Checking Number 621358

Modulus 11 is more difficult to apply and therefore is not as generally used as is modulus 10.

d. Zero Balancing. This technique is associated with batch controls and serves as an input control. Each detail item is accumulated and the final total is subtracted from the batch total. The result is a zero balance if both are correct, hence the name zero balance. This technique can also be applied to tie different computer runs together by assuring the integrity of data and the propriety of machine runs.

e. Compatibility Tests. A compatibility test is a logical comparison between two related fields that can be used to control the input of data or assure adherence to accounting policies. An example of this control as it may be applied to input is a name and soundex code combination. This same technique of associating compatible combinations could relate current date to discount date, or current balance to credit limit, or stock level to reorder point. An invalid comparison of any of the combinations would indicate either an input error or a violation of accounting policy.

f. Limit Check. These checks are sometimes called tolerance checks, and can provide input, housekeeping, and accounting control. Limit checks are programmed into systems to call attention to any data exceeding on less than certain preset limits. Any abnormal items would be displayed for required action by management (management by exception). A check to see that hours worked did not exceed 24 for any one day would be an input edit control, while a similar check to see that hours worked did not exceed 8 hours without some administrative approval would be an accounting control. Limits could be established to check the accuracy of processing by printing out for review the name of any hourly-rate employee receiving a semi-monthly paycheck in excess of $500. It is possible to build multiple levels of limit checks to identify potential or actual errors.

g. Checkpoint and Restart. This processing technique makes it possible, in the event of a hardware error, to localize the time frame at which the error occurred, and return to the last point before the error check point) and resume processing with no loss of data. At this point, the program is restarted and an attempt is made to reconstruct the processing. This type of control aids in the identification of errors and also saves valuable computer time. It also controls the computer processing.

h. Tape Labels. Tape labels control machine procedures by identifying the file and accounting for processing. Labels also control accounting policy by providing for an adequate number of cycles retention for master files. there are two types of tape labels, header labels and trailer labels.

(1) Header Label: This label is the first record on each tape file and should contain a unique identification number, the reel or volume number, the date the file was created, and a purge date. System software is written in such a way that before processing is begun on the contents of the file, the header label is read into the machine and compared with information from the problem program which identifies the file. The header label will also assist in preventing writing to a reel that already contains valid data.

(2) Trailer Label: The last record of each tape file is the trailer. This label contains control totals for the file. Examples of typical tape label formats are shown in figures 3-27 and 3-28.

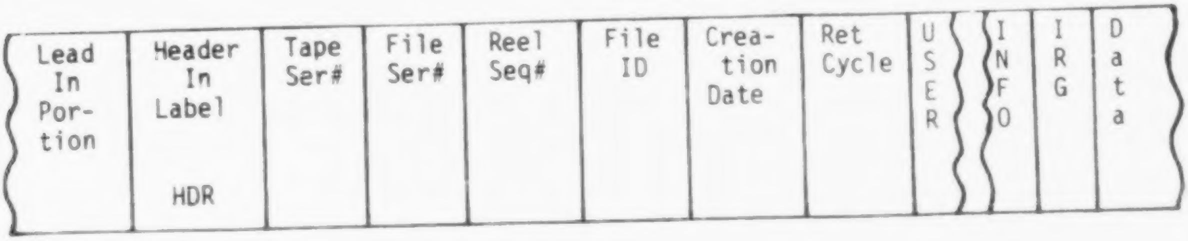

Lead In Portion
Header In Label
HDR
Tape Ser#
File Ser#
Reel Seq#
File ID
Creation Date
Ret Cycle
USER
INFO
IRG
Data

Figure 3-27.

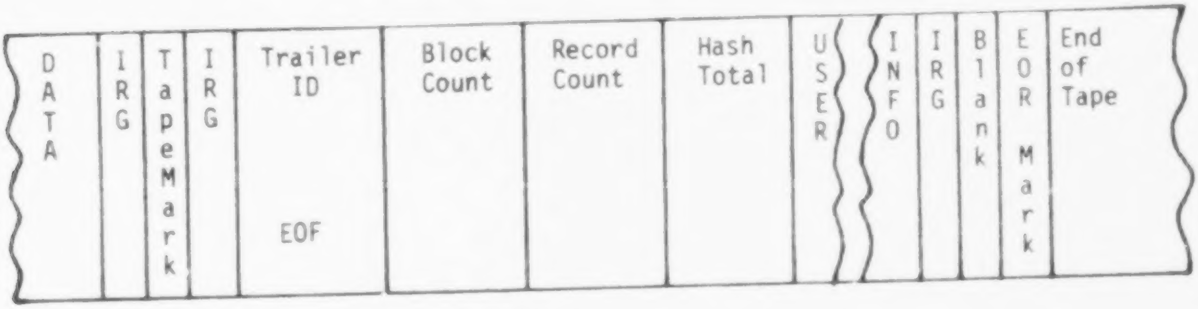

DATA
IRG
TapeMark
IRG
Trailer ID
EOF
Block Count
Record Count
Hash Total
USER
INFO
IRG
Blank
EOR Mark
End of Tape

Figure 3-28.

i. Floating Dollar Sign and Asterisk Protection. In print operations, the floating dollar sign causes the insertion of a dollar sign in the position to the left of the first significant digit of a money amount field. The asterisk protection causes asterisks to appear to the left of significant digits.

3.33. ORGANIZATION CONTROLS. The ADP department is somewhat mysterious with its unfamiliar terminology and unrecognizable machines. With this type of environment and with the concentration of processing within the DP center, there is an excellent atmosphere for undetected mistakes. Consequently, there must be procedures to clarify and ease the relationships between DP and the users, and there must be a plan of organization to control the personnel within the DP center.

a. Control Group. Data processing organizations are usually centralized and should function as a service center whose mission is to serve all users. This type of organization places the data processing department on the organization chart as a service function. There have often been communication problems between data processors and the users, and one way to alleviate these problems is through the use of a control group. The control group is an organization element that provides a buffer between data processing and all the users. The group should be organized independently of both the user activities and the data processing center, but it may be effective as an element of data processing. The control group should perform the followiig functions:

(1) Serve as the primary liaison between using activities and data processors during system design and everyday operations.

(2) Monitor and direct data entering and leaving the data processing center. In this regard, the control group should have primary responsibility for insuring that information flows through the data processing center in the manner outlined in procedures manuals and computer programs.

(3) In some instances, the control group controls the scheduling of the work performed in the data processing center.

(4) The control group's error correction function is of great importance.

(a) The control group should monitor all error correction; i.e., keypunch errors returned to keypunch for correction, and source document errors returned to users for correction. In some cases, certain obvious errors in source documents may be corrected by the control group.

(b) The control group should maintain suspense files containing the errors awaiting corrective action.

(c) The control group can assist the commander in management by exception by maintaining statistical data relative to error frequency and the nature of various errors. This data should prove to be useful for management decisions relating to corrective actions necessitated by frequent errors.

(d) By insuring that input is submitted according to established standards, and by insuring that outputs are correct and in the proper format, the control group can maintain quality control over the data processing operations and products.

b. Separation of Duties: A plan of organization adapted within a data processing facility whereby functions concerned with systems planning and implementation (systems analysts and programmers) are segregated from the functions concerned with the day-to-day operation of the system (machine operators). This separation provides an effective cross-check of the accuracy and propriety of changes introduced into the system; it avoids the undesirable situation in which operating personnel can implement revisions without prior approval and thorough checking; and, finally, it eliminates access to the equipment by non-operating personnel. While the need to adhere to this concept is familiar within a manual system environment, it is at least of equal importance in an ADP system environment due to centralization and concentration of records.

3.24. PROGRAMMING CONTROLS. Programming is a very detailed and exacting process. Consequently errors may occur in processing logic or improper program modification. These areas can be adequately controlled by testing and documenting programs and all changes to them.

a. Testing. Prior to implementing a program, it should be tested for accuracy of processing and for ability to detect and handle all possible types of errors. This testing can be accomplished with test files and parallel processing. The test files should include all possible errors or [illegible] coding, while parallel processing would test all aspects of day-to-day operation in comparison with the old system. This parallel testing should continue for at least three months if possible.

b. Documentation. Documentation consists of documents which [illegible] describe the system and procedures for performing a data processing task. In cases where documentation has been maintained, savings have been noted [illegible] considerable expense when it has become necessary to modify programs. The computer run is the basic unit on which documentation is based, but because of the interrelationship of computer runs, an application overview which shows how the different runs are used is necessary for complete understanding of the runs. This system overview (including the system flowchart) may be included in the various run documentation manuals. The run manual is prepared by the systems analyst and the programmers. It contains a complete description of the programs used for a particular data processing run. Run manuals should contain at least the following:

(1) Problem definition (project background, request, and problem statement).

(2) System description (system overview).

(3) Program description (decision tables, program listings, report layouts).

(4) Operating instructions.

(5) Listing of controls.

(6) Acceptance record (tests and approvals).

Further detail in regard to systems documentation is provided in Chapter 4 of this text.

c. Program Maintenance. There should be a record of all changes made to a program since its original implementation. The program change record is useful in keeping documentation current. It is very time-consuming to correct all documentation, including flowcharts, each time a program change is made. Instead, a change record is included with the rest of the documentation and, unless there is a rather substantial change to the system, flowcharts are not redrawn nor is the other documentation rewritten. The original documentation plus the change record provide the current system documentation. The change record should show, as a minimum:

(1) Date of change.

(2) Reason for change.

(3) Approval of change initiation.

(4) Approval of change implementation.

3.35. OPERATING CONTROLS. Regardless of the accuracy of the data, the computer, and the programs, an operator error may negate all other control provisions. An operator may introduce errors into the system by incorrect machine setup procedures. Operating controls attempt to reduce the possibility of such errors by providing explicit instructions to the operator and by implementing protective measures.

a. Operator Instructions. The set on computer operator instructions (also called a run book) is a reproduction of one section of the documentation. The concept of separation of duties implies that the computer operator does not have access to the complete documentation. Without the complete run documentation it would be difficult for the computer operator to alter a program for unauthorized purposes. The run book should, as a minimum, contain:

(1) Program identification.

(2) Card layouts and keypunch instructions.

(3) Machine setup and take-down instructions.

(4) Input setup.

(5) Switch settings.

(6) Operator duties.

(7) Console messages, responses, and halts.

(8) End-of-job instructions.

b. Console record. A prenumbered, printed record of all the external manipulations of a computer performed by the operator. It is important that the system be so designed that significant steps in each program (control balances, label checks, operator options, and so on), as well as all decisions made by computer operators are printed out on the console typewriter. Thus, a continuous record is provided of all processing steps and of all decisions made. The console record is therefore an eFfective administrative and auditing tool and should be carefully reviewed and controlled by the supervisor.

c. Storage procedures. A system of data retention whereby source documents, punched cards, magnetic tapes, and magnetic disks are filed in such a manner as to safeguard against misuse or inadvertent handling and fire hazards. Significant control procedures for storage are:

(1) Strict control over the access to the tape or disk library (vault). This is related to the procedural controls which are required over changes to the instruction tape and are necessaay to prevent unauthorized individuals from obtaining tapes, thereby circumventing the other controls which have been established. In general, practices should be established which limit access to the tape library to only those people who require tapes for use in connection with the normal operation of the system. In addition, duplicate files should be maintained for all master files and computer programs.

(2) In file processing, magnetic tapes containing master files should never be reused until at least two succeeding versions of the master file have been created. Assuming daily processing, this requires the preservation of three successive days output tapes, together with corresponding transaction data. The idea of keeping three generations of a file is known as the "grandfather-father-son" concept.

d. External Labels. Files should be clearly labeled so that an operator can be sure of their contents. External labeling can be used on punched card files, magnetic tapes and diskettes. It is also possible to use the color of plastic reel jackets for identification. An installation may work out an identification code, such as red for master files, yellow for system files, green for list files, etc.

e. File Protection Rings. Plastic rings which fit into a groove on the back of tape reels are called file protection rings. Once a ring has been mounted on a tape reel, the reel can be either read from or written to, but no writing can be done to a reel if the ring has been removed.

f. Backup Machine. The data processing center should have a backup agreement with some other organization that has a compatible machine. This agreement will insure continuity of operations in the event of a mechanical breakdown or any unforeseen emergency. Realizing that all runs cannot be made as in normal processing, priorities should be established allowing the most significant runs to be processed on the backup equipment.

3.36. SUMMARY. a. Although the objectives of internal control remain the same in ADP as in manual systems, the means of achieving these objectives are somewhat different. The techniques used to control an ADP system fall into categories which reflect the ADP environment:

(1) Input - creation, conversion, and movement of data.

(2) Output - validity of output.

(3) Hardware - electromechanical reliability of the computer.

(4) Programmed - program instructions as substitutes for human awareness.

(5) Organization - proper lines of reponsibility.

(6) Programming - organized approach to program development.

(7) Operating - explicit procedures and measures over computer operation.

b. A balanced system of internal control in an ADP environment must exist to insure its integrity and enhance its economic feasibility. In the final analysis, the adequacy of the network of control present in a system is the key element to be depended upon in determining the reliance the user can place upon the accuracy of the data processing system.

Section 6 - INTRODUCTION TO TELEPROCESSING SYSTEMS DESIGN

3.37. INTRODUCTION. Since the conception of automated data processing, a vast improvement or movement forward in the state-of-the-art has taken place. Great amounts of information are being processed at unimaginable speeds. Yet, what has happened in the area of getting information to and from the computer site? Up to this point, transmission of information from one point to another has been slow at its best, relying on the mail or on couriers. The time has arrived when these methods are no longer satisfactory. In the future, more and more systems will employ terminals allowing the user to communicate directly with a computer located elsewhere. The computer exists for only one purpose and that purpose is to serve man. That purpose can only be satisfied if the information is provided to the user in time to influence a decision. The technique of teleprocessing is an answer to the problem of getting the information to where it is needed when it is needed.

3.38. CHARACTERISTICS OF TELEPROCESSING SYSTEMS: Regardless of the type of application being supported, four general characteristics tend to apply in all cases. These characteristics are really the result of the nature of the operation or the environment in which the system is to function. The environment is one where customers or users are on-line with a host computer via a terminal. There may or may not be any structured schedule where each user has a set period of time to use the system. Furthermore, because of the requirements imposed on the system, the hardware and the operating system must be capable of operating in a multiprogramming situation.

a. Large Scale. Normally, this type of system will require a medium to large scale computer with a vast internal memory as the host computer. Since the system will be employing on-line concepts, the files of the systems being supported will need to be organized in a manner permitting fast retrieval. This will require direct access storage devices. The total required will depend on the number and size of files which need to be on-line at the host computer end of the system.

b. Complex Equipment. Special and complex equipment, other than what one would come in contact with in a normal computer operation, will be employed, for example, the communications line itself is available in a wide spectrum of grades and conditions. The problem of interfacing the data processing equipment with communications lines requires additional hardware which is also available in a wide variety. Selection depends on the functions to be performed by other hardware, and on the communications lines used. The configuration of the terminals depends on the needs of customers and is only restricted by these needs. Any I/O device can be used at the terminal end, just as it can be used at the computer end. Now everything must work together, from the host computer through the communications lines to the terminal producing the end product.

c. Complex Software. Keeping in mind what has been discussed regarding these characteristics of the teleprocessing system and the environment, it is easy to see that more than a standard operating system is required. An additional control program must act as the interface between the standard operating system and the users. Something must be there to receive, interpret, act upon a user's request, and then distribute the result to the right terminal.

d. High Degree of Systems Reliability. In a batch type environment, it is a simple matter to distribute the output to the right customer, or to determine that a customer is, in fact, a valid customer with access rights to the information. But what about a situation in which a user will be serviced via an on-line terminal? Now the system must keep tract. Exact accounting procedures must be maintained to preserve the security of the system. No longer will a human decide the authenticity of the user's request. The system must first determine if the requestor is a legitimate user. The fact that the requestor has been determined to be a legitimate user does not necessarily mean that he is to have free access to all information. The information must be "protected" so that only authorized user can gain access. Having generated this output, the system is responsible for still one other step, and that is to see that the information is distributed to the right user.

3.39. MANAGEMENT-DEFINED GOALS. As already discussed in previous chapters, management is responsible for establishing the goals and objectives of any new system. When dealing with a teleprocessing system, there are three specific goals which must be established, and which can only be established by management.

a. Response Time. Response time is best defined as the time interval between the operator's pressing the last key of a transmission and the terminal's typing of the first letter of the reply. Only the user can specify how fast the system must react to an initiated inquiry. When that information has been provided, the analyst is interested in designing a system which will provide the necessary information in time to influence the decision-making process. If the system can do that, it is meeting the specified response time. A critical response time for one terminal may be seconds, where in others, it may be hours. For example, one of the terminals may be a reactor, and the host computer is contolling the critical mass. Here we could well be talking about fractions of a second for a response time.

b. Systems Accuracy. The amount of faitah a user has in a system depends on the accuracy of the information within that system. When dealing with accuracy, the systems analyst mut deal with degrees, or with a goal expressed quantitatively. The degree or percentage of accuracy within a system can only be established by management. It is up to the system analyst to employ the necessary controls to meet the standards. At the same time, management must be realistic when quantifying accuracy. Management may not fully understand what a zero error rate will cost. It seems that as the percent of accuracy increases on a linear basis, the cost of hardware increases exponentially.

c. Systems Availability. The last goal which must be established by management is system availability. Response time is the time lapse between the inquiry and the response. System availability is the elapsed time in a "wait" state, or when the system answers with a busy signal. Again, only management can specify what will be a tolerable wait time. The systems analyst must then design the system by employing a variety of techniques and hardware peculiar to teleprocessing to provide adequate user support. As in response time, each user may have different requirements regarding system availability. Each user must be considered as an entity when designing the new system.

3.40. ANALYST-DEFINED PARAMETERS. When interviewing the users of the new system, the systems analyst is interested in additional information which, like the management-defined goals, will affect the design of the new system. The systems analyst will still be designing the new output to meet the user's requirements and the inputs needed to get the information into the system, but he must go beyond that, and examine the following items as well.

a. Distribution Pattern. The analyst must consider the distribution of the users from two aspects - their location, and the priority of their data. In both cases, the number of users must be considered, not the number of terminals that will be required. In the end product, it may be that several customers will be serviced by the same terminal, but that will probably be an unknown at this stage of the design effort.

(1) Location. Determine the physical location of each user in relation to the host computer and to other users. The best way is simply to take a map and plot the locations by using map pins. An analysis can then be made as to the distance between customers or groups of customers; the number of different physical locations; whether the users tend to be in groups or to form clusters; or if they follow a path. The systems analyst needs this kind of knowledge when he begins to develop the communication network.

(2) Priority. Among the users, one may require closer attention than the others within the network. If this is the case, that user may require service more often, or may need to be recognized immediately when he requests service. As an example, one terminal may be a nuclear reactor with the host computer controlling its critical mass. Another terminal may be supporting an inventory control process. In this case, it is apparent that

more attention muist be given to the reactor, and that it should not be required to wait. The individual putting information into the inventory control process could wait and his operation would not suffer from it. However, there could be an adverse effect if the reactor were ignored, or if it was forced to wait for any period of time.

b. Volume. Another piece of information which must be collected for each of the users is the volume of data to be sent and received. When collecting the data, it is necessary to collect both the average and peak volumes. Each will have a different effect on the final product. The question which must be dealt with is whether the system should be designed to handle the average flow, the peak flow, or some amount in between. One other bit of information relating to the peak volume is needed before a decision can be made - what is the time? When do the peaks occur? If the peaks all occur at the same time and last for a very short span, then maybe some sort of smoothing technique should be employed, or the load should be redistributed. Also, when looking at the volume of traffic, all transmission requirements should be examined. There may be requirements to transmit data between users or between terminals as well as between the normal route of terminals and host computer. Again, if there is a great deal of information exchanged between terminals, this will have a definite bearing on the final network and the types of hardware used. One means of recording the information as it is collected is to construct a matrix. A "Data Distribution Chart" (Figure 3-29) is an example of such a matrix and provides for the recording of flow and volume by user and document or report.

c. Language of the Information. The systems analyst must consider the form or language that the customer will use to correspond with the system and the form or language the system will use to communicate with the user. It may be anything from clear text to a complex coding system. With clear text, it is easier for the user to determine correct verbiage in case of a transmission error. As an example, examine the following sentence:

There is an error in tha sentence you have just read.

In this example of clear text, it would be an easy matter to correct the misspelling without asking for a retransmission of the message. Clear text, however, will result in an increase in the number of characters transmitted, which leads to increased costs. Normally the number of characters being transmitted will increase as teachniques are employed to make the system easier to use.

DATA DISTRIBUTION CHART						
TO FROM						
	REPORT	VOLUME	REPORT	VOLUME	REPORT	VOLUME

Figure 3-29.

3.41. SELECTION OF DATA TERMINALS. The customer uses the system by means of a terminal. This is his primary, if not his only, point of contact with the system. The terminal has a great deal to do with first impressions on the part of the user, and has a bearing on how much he will use the system. The overall operation and efficiency of a system can be seriously affected by the terminal configuration. A poorly planned terminal configuration can have an adverse effect on run time, error rate, cost, and usefulness. Care must be exercised during the selection process to insure that the proper type of equipment is selected to fill the needs of the customers and to satisfy the goals and constraints established by management. Particular attention must be paid to four specific areas since there is a wide variety of equipment available which could be used as terminals.

a. Cost. Since a wide variety of available equipment exists, the cost for this equipment also varies. At one end of the scale, a terminal could be nothing more than a teletype (Figure 3-30) while at the other end of the scale, the terminal could be programmable - another computer (Figure 3-31). In fact, most computer I/O devices can be used as terminals. Even within a particular type of device, such as a cathode ray tube (CRT), a variety is available. The selection and cost of the terminal will depend on what is is expected to do.

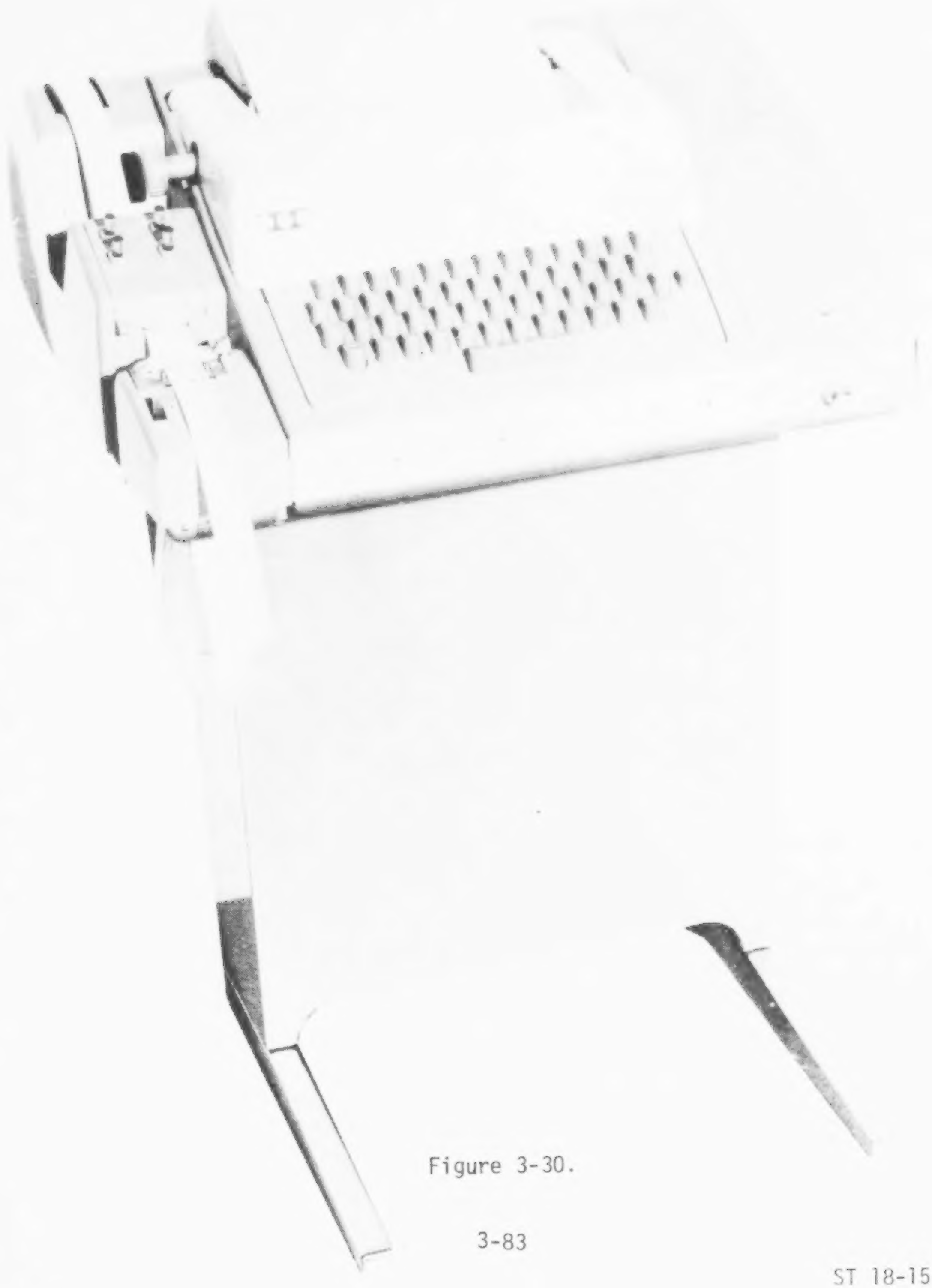


Figure 3-30.

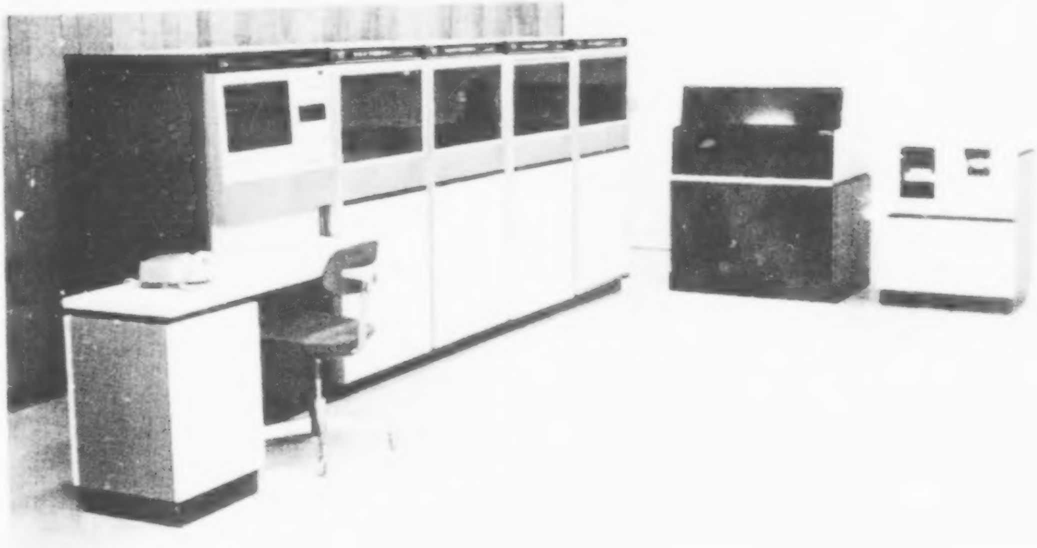


Figure 3-31. Mohawk System 2400

b. I/O Requirements. Since the users are the ones to be satisfied, their I/O requirements must be considered. Their requirements will dictate the format and media of user information, the speed at which it will be processed, and the format and media of the information received by the user. Basically, the four following forms are applicable to I/O requirements.

(1) Transaction-oriented. In a transaction-oriented environment, the information is collected at the point of origin and transmitted at the time the transaction occurs to the host computer to update the file. The file reflects what is happening as it happens. A point-of-sales system is an example of a transaction-oriented environment. The cash registers in the retail stores are on-line terminals. As a sale takes place, the information is keyed in on the cash register. The information on the sale is then transmitted to the host computer where the file is updated to show that the item has been removed from the inventory. In this system, the file reflects what is on hand for any given amount. An extension of this type system is one where the terminal is configured with a light pen and a cash register. This system is keyed on the Universal product code (UPC) (Figure 3- 32) which is affixed to each product. The UPC is read by the light pen at the checkout station, and transmitted to the host computer. The file is then searched,

the proper price retrieved, the balance on hand updated to reflect the change, and the price of the article rung up on the cash register (Figure 3-33). With this type system, there is minimum intervention on the part of the sales clerk. Since the price is stored in the file and not entered, there will be a minimum of pricing errors. Low speed input and output devices are characteristics of this system.

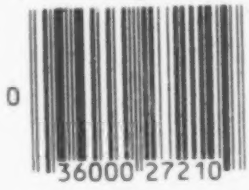


Figure 3-32. Universal Product Code

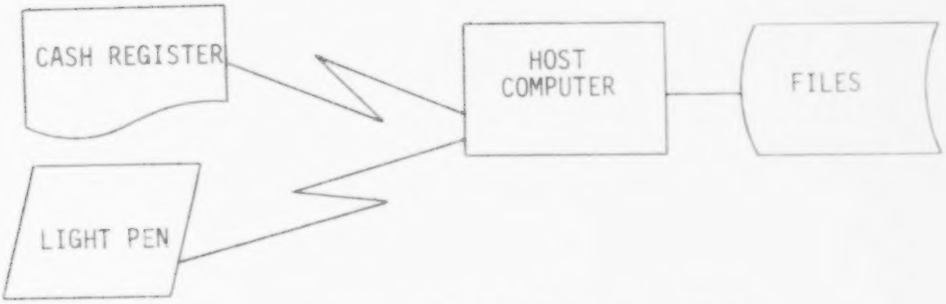


Figure 3-33. Point-of-Sales System.

(2) Batch Input. Some systems do not require the immediate transmission of information. These requirements allow the information to be stored at the terminal end as it is being collected. Normally magnetic tape is used as the storage media, collecting the information as it is being entered through slow speed devices such as by keyboard input. Some time later, the information is transmitted as one message. In a transaction-oriented system the speed of transmission depends upon the speed of the individual who enters the data. A batch input system takes advantage of the speed of the storage media used; and normally transmits an entire message at a time, not just one character at a time. Batch input systems allow the scheduling of time amounts to the users. A user comes on at a set time, transmits all the collected data, and then terminates. Some point-of-sales systems include batch input provisions. As an example, one retail chain employs point-of-sale and batch input in this fashion: as the sale occurs, it is rung up on the cash register (terminal), and the files are updated to reflect the sale; up to this point it is a standard transaction-oriented system. At the same time, the sales information is collected and stored on a direct access storage device. At a specific time in the evening, all

collected data is transmitted to a host computer at the regional headquarters. This part of the system is a batch input operation (Figure 3-34). This type of system is characterized by low-speed input, but fast output. Another example of batch input would be to use a device that has a storage capability. The data could be stored in the terminal as it is keyed in, then, at some later time transmitted to the host computer. An example of such a device is shown in Figure 3-35.

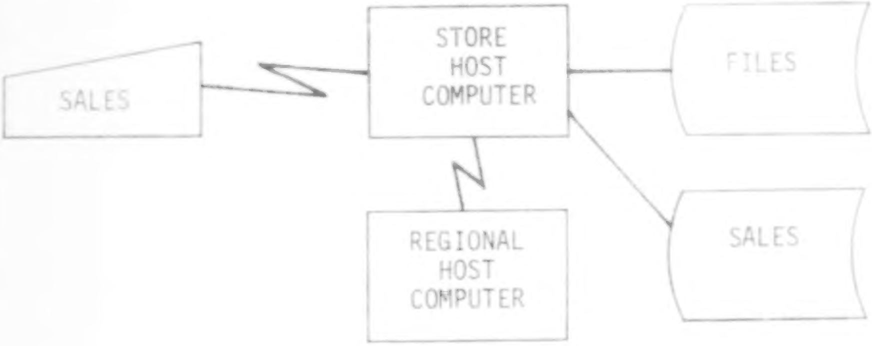

SALES
STORE HOST COMPUTER
FILES
REGIONAL HOST COMPUTER
SALES

Figure 3-34. Batch Input Operation

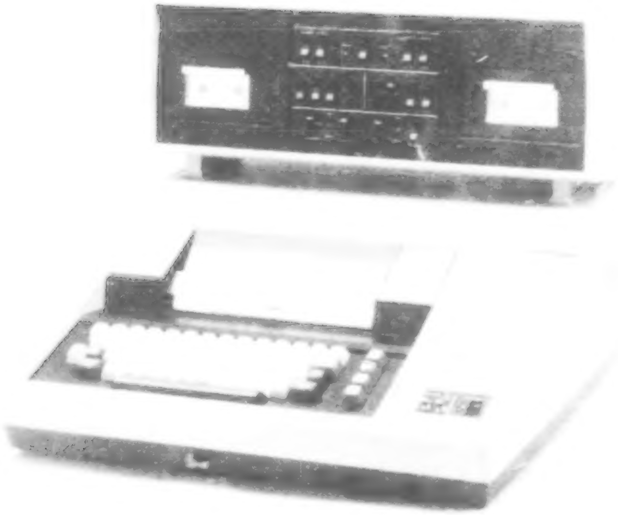


Figure 3-35. Data Terminal
(Texas Instruments Model 742

(3) Batch Output. The program just discussed can be reversed, in which case the host computer transmits a volume of data to the terminal very quickly. However, the terminal in this case needs some type of media on which to store the information. The storage media used can be punched card, magnetic tape, disk, or even hard copy. As in batch input, batch output processing employs collection of the data. Batch input is based on collection of data prior to transmission to the host computer, while batch output is the collection of data at the terminal. In this way, an entire message or report could be transmitted, stored at the terminal on magnetic tape, and after transmission is completed the information could be reproduced on hard copy as an off-line operation. A terminal device such as that shown in Figure 3-38 would support this type of operation. As information comes in, it would be stored on a magnetic tape cassette for later printing. This way, the speed of the network would not be degraded to the speed of printing.

(4) Retrieval System. The last of the four I/O forms considered is the type of application which requires the immediate retrieval of information in answer to a user inquiry. There is a difference between this requirement and a transaction-oriented system. In a transaction-oriented system, the files are updated as an action occurs. With retrieval systems, information is retrieved from existing files. An airline reservation system is an example of a retrieval system. Normally, the terminals are cathode ray tubes (Figure 3-36). The airline representative requests the desired flight information through the keyboard. When the information has been retrieved from the host computer, it is transmited back to the terminal and displayed on the screen. Then the potential pasenger has the necessary information to make a decision. It is of key importance to the system that the information be located very quickly and transmitted to the screen of the terminal. Also, in a strict retrieval system, files are not updated; they are only searched and any desired information is retrieved. A system may end up being a combination. It is possible to have a combined retrieval system and transaction-oriented system. The airline reservation system employs this combination. It must support the retrieval capability when the potential passenger needs to know what flights are available; and it must operate as a transaction-oriented system when the passenger makes a decision and then update the files with his reservation. This type of I/O requirement will have an effect on the host computer configuration as far as the necessary storage facilities for the files are concerned. With a transactionoriented system or a retrieval system, the files will be required to be online at all times while the system is in operation. For example, with the point-of-sale system, the files would have to be on-line during the hours when the store is open for business because there is no way to determine when a sale will take place. Yet, in a batch input system, there is no need for the files to even be on line when the information is being transmitted.

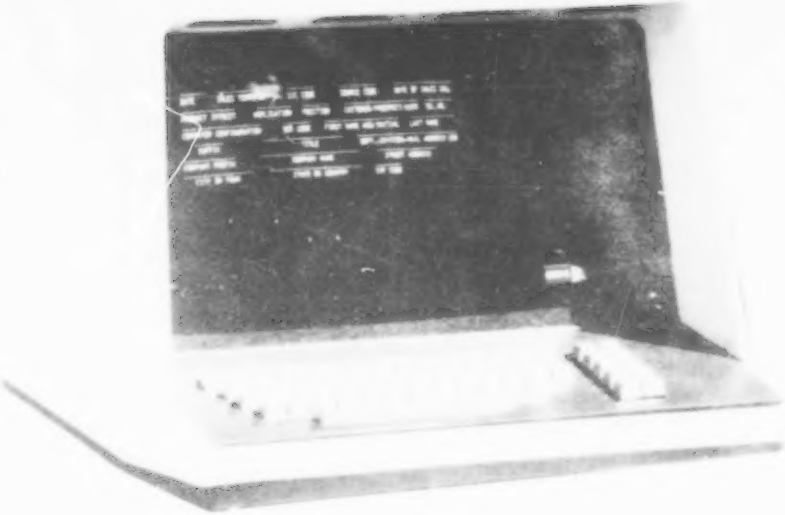


Figure 3-36. CRT (Sanders Model 804)

c. Environment. Another factor that will affect the selection process is the physical environment, or where the terminal will be located. The desired location of the terminal can restrict the types of I/O devices. Some devices, such as the teletype, are rather noisy. This would not be a logical choice for a commander's office. Also, size may be a limiting factor. Maybe the terminal is to be mounted on a vehicle. There is only a limited amount of space available on the vehicle in which to place the terminal. Also, the climatic conditions in which the terminal will be expected to function can be a limiting factor. If the manufacturer's specifications call for a temperature range between 65 and 75 degrees with 50% relative humidity as the operating environment, then that device would not be the proper choice for use in a hot, dusty warehouse unless modifications were made to the selected site. In conclusion, then, three questions must be examined by the systems analyst.

(1) How disruptive will the terminal be to those working in the immediate area? Noise can be a real problem and disrupt those working in the same surrounding area.

(2) What are the climatic conditions in which the terminal will be expected to function? Existing conditions may not support the terminal selected, thereby claling for special site preparation.

(3) How much physical space will be required by the terminal? The chosen terminal must be able to fit into whatever amount of space is available for it at the intended site.

d. User Considerations. The last factor that has an effect on terminal selection is the user. The terminal exists to provide a service by satisfying the user's input and output requirements. In fact, the success or failure of a system may depend on how easily the user is able to communicate with the system through the terminal. This is especially true in those cases where the system is primarily directed toward communicating with the human being. Consequentially, there are some details pertaining to the potential user that must be analyzed by the systems personnel. There are as many types of applications as there are users. The users will have a wide variety of backgrounds and education, from a clerk in a warehouse who has a high-school education, to a researcher who has a Ph.D. in a medical research program. Each will have different needs and requirements; therefore, different type terminals and different levels of training will be required for each. In fact, the situation may involve anyone from a casual user to a full time operator who does not require a background in programming to one who has an extensive programming background.

3.42. BASIC TASKS OF TERMINALS. At this point, when speaking of a "terminal", the term is all-encompassing, beginning with the termination of the communication line. The equipment that will be needed at the terminal end will depend on the basic tasks the terminals will need to support or accomplish. Three basic tasks associated with terminals fall into this category: handling control signals, data input and output, and code translation.

a. Control Signals. Associated with the transmission of data is a signal that denotes the beginning and the end of an entity. Just what constitutes an entity depends on the mode of transmission. The type of control signal also depends on the mode of transmission. The two modes of transmission are synchronous and asynchronous.

(1) Asynchronous. In asynchronous or start/stop mode, transmission of data will take place one character at a time. This is the normal mode for key input devices having no buffering capability. As the key is struck by the operator, a character is transmitted. Assosiated with each character is a strat signal and a stop signal. The start signal is a "0" signal which identifies the beginning of the character, and the stop signal, a "1" condition, terminates the character. Therefore, the bit conditions between the two form the character. Any standard code can be used for transmission.

The character code chosen though, does alter the length of the stop signal. If the 5-bit Baudot code is chosen, the stop signal will be 1.42 elements (Figure 3- 37) but if the ASCII code is chosen, the stop signal will be 2 elements (Figure 3- 38) in length. As a minimum, whatever code structure is chosen, the receiving terminal expects to find that number of bits between the start and stop signals. The receiving end will be alerted to a new character being transmitted when the stop signal (1) is changed to a start (0). In cases dealing with a key-driven input device, there will be cases where the stop signal will be longer than 1.42 elements or 2 elements. The time between key strokes can vary depending on the operator. Since transmission only occurs when a key is struck, there will be pauses in the transmission; and the stop signal will last for the length of the pause. The start signal is constant in length, lasting for one element, and is not affected by the selected code structure or the length of the pauses in transmission.

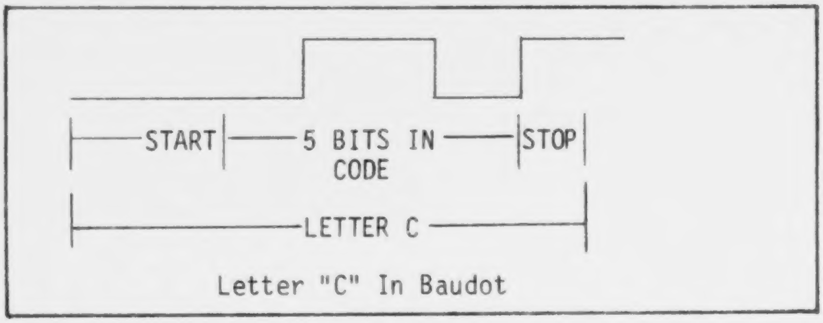


Figure 3-37.

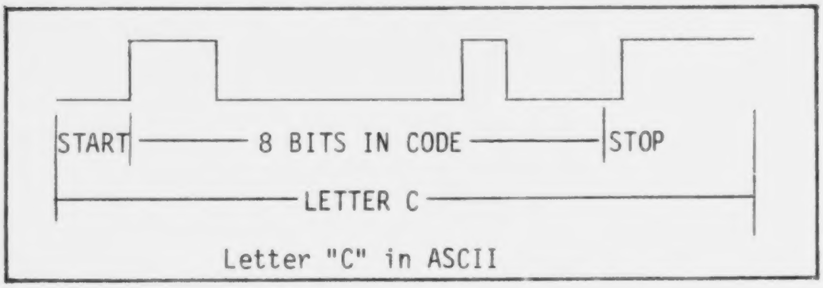


Figure 3-38.-

(2) Synchronous. The second mode of transmission is synchronous. With this method, there is no start and stop signal associated with each character, and transmission of data is not one character at a time. In synchronous transmission, a complete block of data is transmitted at one time within a specified period of time. The block size is independent in

this type of operation and can be any length, normally only restricted by the physical limitations of the I/O hardware. As an example, the block size is normally 80 characters when transmitting punched card format, or 133 characters when transmitting hard copy report format. Although a block of data is being transmitted, the transmission must still be controlled. There is still a control signal, but it is now a synchronization signal. The purpose of the signal is to synchronize the receiving and sending ends of the system. Imagine each end as having an internal clock. Before a block of data is sent, a synchronization signal is sent out by the sending end which synchronizes the two clocks. The hands are now moving at the same speed. If the hand moves from A to B at the sending end to transmit a block of data, the hand moving the same distance at the receiving end signals the receipt of a block of data (Figure 3- 39.

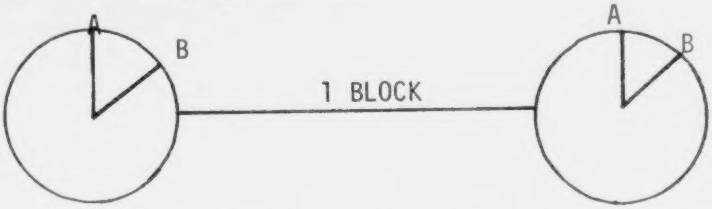


Figure 3-39. Synchronous Transmission

With asynchronous transmission, a character is transmitted independent of any other character in comparison to synchronous transmission where a block of several characters is transmitted at one time. This places a requirement to be able to hold the data until the block is complete. At the same time, synchronous transmission does not have the control signal associated with each character; and the pauses are eliminated. Therefore the percentage of redundant data is lower. Transmission speed is faster with synchronous transmission because the only restriction is the speed of the line. As a result, this mode is more expensive as far as terminals are concerned because the terminal must have buffering capabilities to support synchronous transmission.

b. Data Input/Output. In support of an on-line operation, data will be shifted in and out of the terminal, to and from the host computer. This can be further compounded when there are several I/O devices at a given terminal and if the devices should or should not be buffered. This question can be carried one step further and can be addressing the issue of double buffering. The speed at which the uer must transmit in relation to the type of I/O device will dictate whether that device should be buffered or not. A good example would be a keyboard input device connected to a 4800-bit-per-second line. By going to an asynchronous mode, unbuffered, operation, the network could be degraded to the speed of the operator at the terminal. Buffering capability could be added, allowing the system to support synchronous transmission. Maximum utilization of the network could then be achieved. As the

best copy available

operator prepared the message, it ould be held in a buffer. When the buffer is full, the entire block would be transmitted. In this case, buffering would be beneficial. However, if the I/O device happened to be magnetic tape or a device matched to the speed of the communication line, the need for a buffer would be negated.

c. Code Translation. There are a variety of transmittable codes, from Baudot (5 bit) to EBCDIC (9 bit). The code used most widely for transmission is ASCII. It has been accepted as the standard code structure for teleprocessing. Code conversion may or may not have to take place. The I/O device may be compatible to the code selected for transmission. In the example in Figure 3- 40 ASCII is the selected code, and the I/O device will accept it without need for conversion.

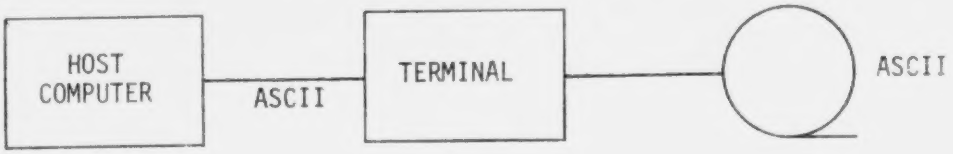


Figure 3-40.

At the terminal end though, there may be several types of I/O devices, each accepting a different code structure. These different codes must be converted to one single code for transmission from the terminal to the host computer (Figure 3- 41). The reverse also applies. When data is received at the terminal in ASCII format, it may require conversion to the specific code for that I/O device.

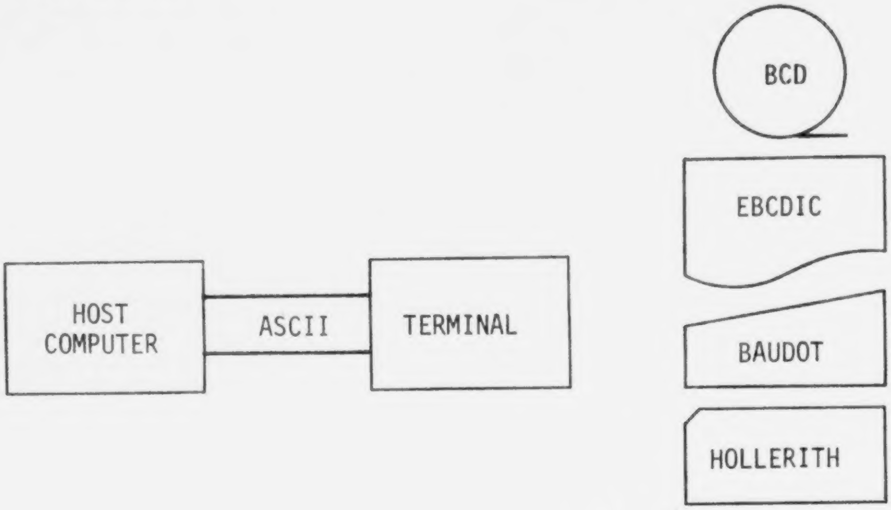


Figure 3-41.

3.43. PROGRAMMABLE TERMINALS. At the top end of the price scale are the programmable terminals, sometimes referred to as intelligent terminals. These terminals came into being with the introduction of minicomputers and microcomputers. A minicomputer with the necessary I/O devices becomes a very powerful and flexible terminal, limited only by the creativity of the programmer. Although expensive in comparison to other terminals, programmable terminals should not be ignored strictly on the basis of their cost. The benefits of programmable terminals may offset their cost, since they can accomplish a number of functions in lieu of dependence on a host computer.

a. I/O Formatting. If the user at a terminal requests a particular report through a non-programmable terminal, the entire report will be formatted at the host computer. All the headings must be transmitted as well as each line of the report, just as it should appear, to include the proper spacing. This takes a lot of time on the part of the host computer and in transmitting the data. Why not pre-position the heading information at the terminal? This could be done with a programmable terminal. In fact, with this type terminal, only the new data need be transmitted. The report would be formatted at the terminal. The host computer would be tied up for less time, and less data would need be transmitted. A programmable terminal can also assist the user in preparing the input, especially if that input must be in a standard format. The format could be displayed on a CRT, and the user need only fill in the blanks or provide answers to questions.

b. Data Compression. time and again there will be instancaes where blanks will be present in data to be transmitted. As an example, maybe a particular job calls for input by punched card where only the first 50 columns will contain data while the last 30 will always be blank. Going back to a non-programmable terminal with a card reader, the entire card (80 columns) will be transmitted - 30 blank characters for each. With a programmable terminal, these blanks can be stripped off and only the significant characters transmitted. At the receiving end it is a simple matter to treat the incoming blocks as 80-character blocks although only 50 characters were sent forward. There will also be times when standard messages apply. In these cases, there is really no need to send the entire message. A code can be sent; and upon receipt, the code can be translated to the message. Other examples could be cited but, basically, the purpose behind data compression is to send only significant data that eliminates as many blanks as possible. It takes just as much time and costs just as much to send a blank as it does to send any other character.

c. Error Detection. Any time humans prepare input, erors can and will occur. If a programmable terminal is available, the data can be edited and only information passing the edits is transceived. This is especially important when there is a high error rate. Catching errors at the sending end results in several benefits. First and foremost is the fact that transmission of erroneous data is eliminated. When an error is caught in input, it can be turned around more quickly for the user to make corrections. If editing is accomplished by the host computer, all data, both good bad, is transmitted. The host computer is then tied up accomplishing the editing. Any errors found would have to be sent back to the terminal for correction. Once corrections had been made, the data would again be transmitted to the host computer. This all takes time, ties up the host computer on tasks that could have been performed by the terminal, delays the actual processing, and increases the cost. Also, it is possible that the data being received by the terminal can be in error. Advantage can be taken of the terminal's ability to be programmed, and edits can be written to catch the errors. If data is supposed to be numberic, or contains a certain number of characters, it can be checked for these types of discrepancies.

d. Data Collection. In situations where input devices are character-by-character oriented (key input), the programmable terminal becomes very effective. It can be used to collect data as it is being generated at the keyboard. When the terminal has a complete block or message, it is transmitted. Use is made of the terminal's buffering capability to reduce transmission time by going from character-by-character (asynchronous) transmission to block or message transmission (synchronous). In some situations, several input stations may be at the terminal end. The programmable terminal is then responsible for collecting the data from each input station. It is actually a gathering-place supporting multiple input stations that are remote from a host computer. A separate storage area can be set aside for each input station. As information is generated it is stored in the area reserved for the I/O device concerned. When the work at the terminal is completed, the data which has been collected can be transmitted.

e. Automatic Restart. At times during transmission, the flow of data betwen the host computer and the terminal will be interrupted or the system will malfunction. By keeping a log of the transactions processed through the terminal and taking periodic checkpoints, a record can be kept to determine what is happening or to find the condition of the programmable terminal at set times. If a malfunction or interruption occurs, the terminal can be restarted at the last checkpoint and automatic retransmission of data can be resumed from that point up to the time of the malfunction. This, of course, is a great deal more advantageous than retransmitting the entire message or manually trying to locate the point in the data where retransmission should begin. Either way would be quite costly.

f. Local Support. Some jobs or parts of jobs could possible be accomplished at the terminal end. For instance, a job may call for a price extension (unit price X quantity) to be reflected on a report along with the basic information. The terminal could perform the calculation rather than the host computer. There could be times when the terminal has enough memory available to run the job, thereby eliminating the need to go to the host computer at all. How much can be done at the terminal will depend upon the size and types of I/O devices. But, in any case, a job accomplished at the terminal relieves the host computer and lessens the burden on the network.

g. Fail Soft. This benefit is only apparent when a malfunction of the network occurs. If the terminal were nothing more than a teletype machine, it would be completely out of business. But if the terminal were programmable, some support could be provided. For instance, the data collection function could still be supported. The data could still be edited and the errors corrected. With a programmable terminal, the user would not be completely without support, as would be the case with a non-programmable terminal. Since the programmable terminal is a computer, some support can be provided, but certainly not as much support as if the network were functioning and the host computer were available.

3.44. THE COMMUNICATIONS FACILITIES. Information is transferred between the host computer and the terminal by means of a communications facility. A myriad of corporations are providing a variety of communications facilities on the market today. The forerunner of the industry, and the one that has had the greatest effect is American Telephone and Telegraph Company (AT&T). Transmission of data in support of data processing was a natural step for this commission of data in support of data processing was a natural step for this company since they had the facilities already available through the telephone lines that covered the country. Western Union and Data Transmission Company are two other companies in the market. For east of identification, the communications facilities available from carriers fall within three categories, and are capable of transmission in one of three ways.

a. Classes of Communications Facilities:

(1) Narrowband. Facilities capable of data transmission up to 300 bits per second.

(2) Voiceband. Facilities capable of data transmission from 300 to 9600 bits per second.

(3) Broadband. Facilities capable of data transmission at speeds greater than voiceband. Facilities currently available are capable of data transmission at speeds up to several million bits per second.

b. Transmission Flow.

(1) Simplex. Transmission of data in one direction only is handled by a simplex line (Figure 3-42). This type of transmission would be used in a control process where, for example, a host computer might be in control of a steelrolling process. Along the processing line there would be several

thermometers acting as on-line terminals to the host computer. They would record the temperature of the steel as it passed throught their station, and transmit the information to the computer. There would be no communication from the computer to the thermometers. Since the requirement is for one-way transmission only, a simplex line is all that is needed.

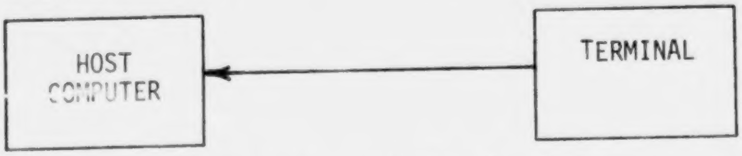


Figure 3-42 . Simplex Service.

(2) Half-Duplex. This type of line will handle transmission of data in two directions, but not at the same time. The terminal is able to send data to the host computer, and the computer can send data to the terminal, but one must wait until the other is finished, because data will only flow one way at a time. The best example of half-duplex (Figure 3-43) is the push-to-talk communications system as used in the MARS network. One person talks, then a physical switch is made so the other person can speak. Half duplex is as if there were a single-lane bridge across a river. Traffic from one side can cross, while traffic on the other side waits. Then the traffic flow is reversed. Thus, half-duplex only requires two wire service.

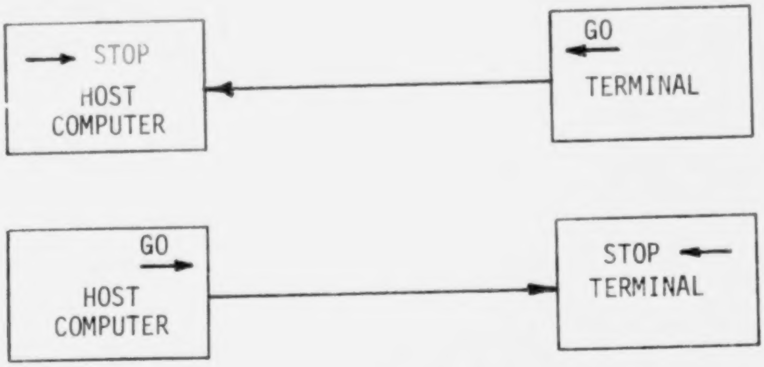


Figure 3-43. Half-Duplex Service.

(3) Full Duplex. This type of line will handle transmission of data in both directions at the same time. The host computer can send data to a terminal and the terminal can send data to the host computer at the same time. There are advantages over half-duplex to this method. The most evident is that one does not have to wait for the other to finish before

data can be transmitted. Full duplex (Figure 3-44) is a very easy way with which to handle the control signal. Data can be sent one way while the control signal is being sent in the other direction. Other uses will be discussed under methods of error detection and correction. Full duplex requires full four-wire service.

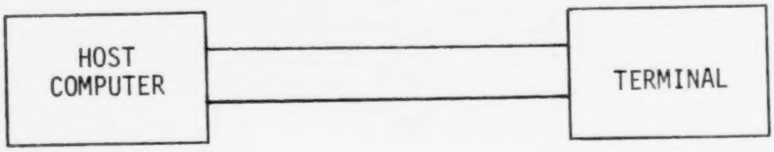


Figure 3-44 . Full Duplex Service.

3.45. NETWORK DESIGN. Basically, each network is the same in that each contains the same type of hardware. There are extensions of the basic network scheme. A basic network will, as a minimum, be composed of a host computer, telecommunications control unit (TCU), Modulator/Demodulator Unit (MODEM), communications line, a second MODEM, and a terminal (Figure 3-45). Terminals and communications lines have already been discussed. At this point the basic line configurations and characteristics will be explained. It is assumed that all networks will contain at least the basic components.

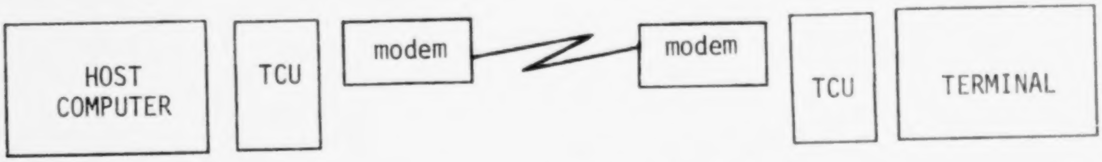


Figure 3-45 Basic Network.

The purpose of each component will be included later in this section. Looking at the schematic in Figure 3-45, you can see that there is not much of a problem as far as the configuration and the relationship of host computer and terminal are concerned. Only one terminal is used. But what if additional terminals were added? Then the problem would become more acute. Would a centralized or a distributed network be better? Can a dial-up line suffice or should it be dedicated? The goals of management and parameters established by the systems analyst now play an all-important part in the design effort.

a. Centralized Versus Distributed Networks. There are two basic network designs, centralized and distributed. Each has benefits and shortcomings; therefore, each must be examined in light of what it will offer and what it must support.

(1) Centralized Network. A centralized system has only one host computer (Figure 3-46). The host computer is responsible for supporting all the terminals in the network. The data base is also centralized. Any terminal desiring information must request this information from the one host computer. Since only one host location is involved, design is very simple. But, since there is only one host site, the system does not have any backup capability. This may or may not be a problem. What must be considered is the effect on the user if the system malfunctions. Backup can be very expensive. It might mean a spare standing by in case the prime system fails. But hthen, how much of a spare? Should it be nothing more than a MODEM, a mainframe, or an entire system?

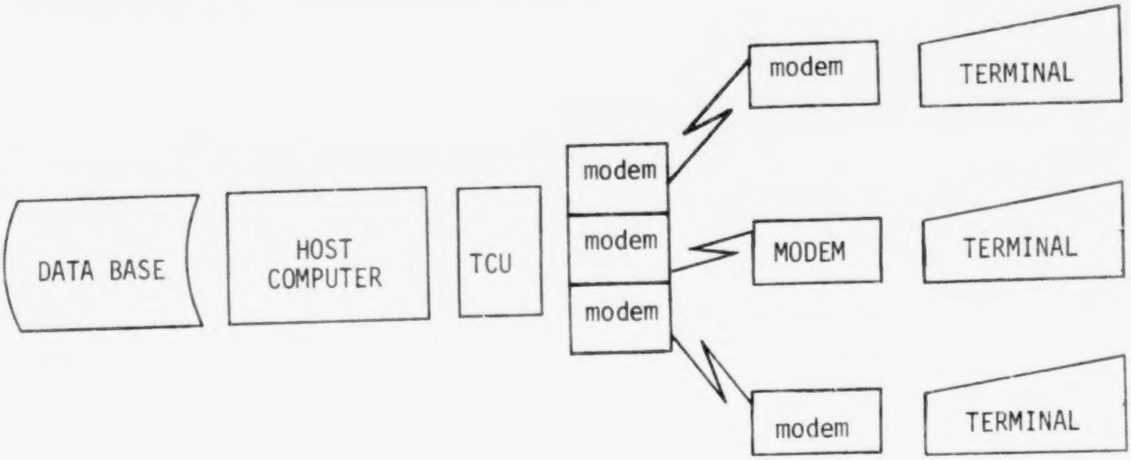


Figure 3-46. Centralized Network.

(2) Distributed Network. Characteristics of this type of network is the use of multiple host computers and distribution of support (Figure 3-47). Normally, the terminals will be segregated along some natural boundaries; i.e., geographical lines, type of support required, etc. A host computer is then required to support a group of terminals, and all the computing power is no longer in one location. Along the same lines, the data base will also be distributed. A distributed system does add problems in design since there is more than one host computer. The computer sites must be able to communicate with each other in order to share information. This is most apparent when distribution is simply along geographical lines, and each terminal has the same type of requirement. A terminal serviced by one host computer may need information that is within the normal responsibility of another host compuer. Provisions must be made to provide the requested

information to the proper terminal. One major benefit of installing a distributed system is the capability of providing backup. Since there are multiple host computers, why not designate one host computer to act as backup for another? For instance, using the network in Figure 3-47 computer "B" could be designated as backup for compter "A", "C" the backup for "B", and "A" as backup for "C". Then, if host computer were to go down, the terminals which normally were serviced by it would be supported by computer "B". There may be some degradation of service to the terminals of both "A" and "B" while this happens, but still, all the terminals are receiving some support. To support this operational scheme, backup will require additional distribution of the data bases, with each computer housing not only its own data base, but also the data base of the computer being backed up. As we have said, host computer "A" is backed up by computer "B", therefore, data base "A" will have to be stored at computer "B" as well as at computer "A".

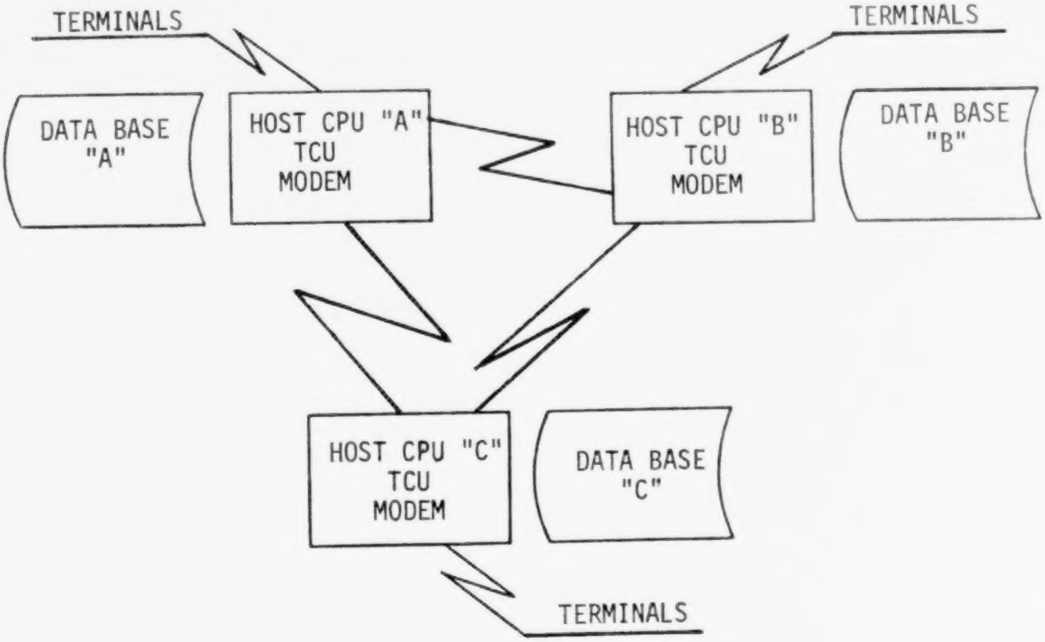


Figure 3-47 Distributed Network.

Now the data base is located at two different sites, both copies of the data base must be maintained and updated. When changes are posted to the data base at computer "A", they must also be posted to the copy of the data base located at computer "B", but how? Should the backup file be updated at the same time as the file at the primary site? A better method would be to do the update of the file at the primary site, and accumulate the changes for later transmission to the backup site where a batch-mode update of the files can be performed. Backup of files can be handled in a variety of ways, but the basic fact remains that all copies of the files will require updating to

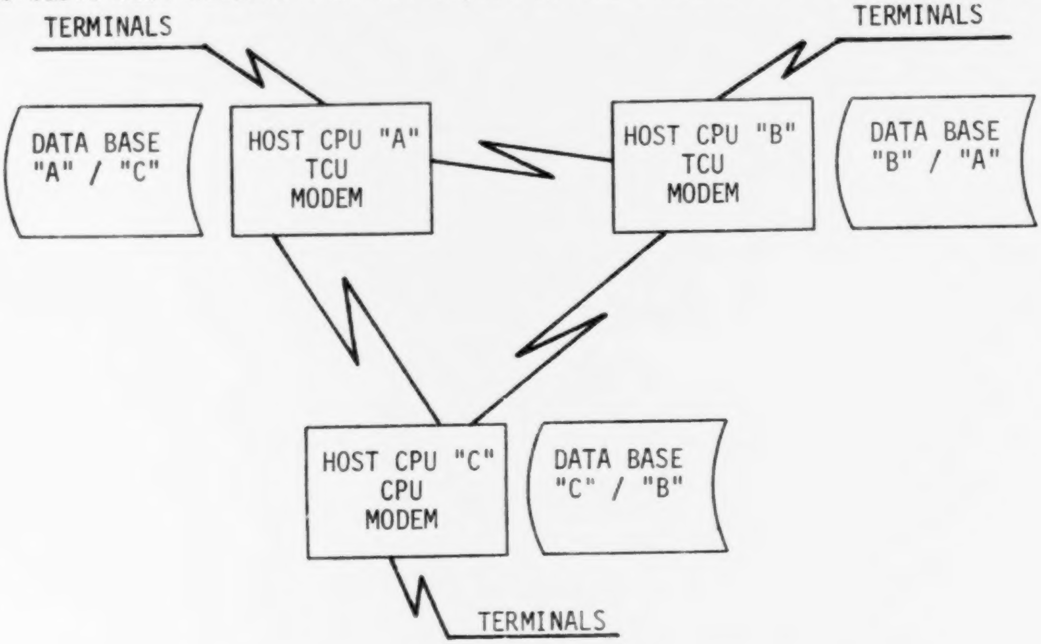


Figure 3-48.. Distributed Network With Backup

fully support the backup capability. Obviously, there are some drawbacks to a distributed system. Among these are cost and security. Increased costs are self-evident. The system will employ multiple host computers, and there will be costs for the communications lines between them. The problem of security will be greater simply because the data bases will be in more than one site. Whenever multiple copies of a data base exist, the chances of compromise increase. Figure 3-48 illustrates a distributed network system with backup capabilities.

b. Dial-up Service Versus Dedicated Service. Through public channels, two types of service are available: dial-up and dedicated. As with anything where a choice exists, there are benefits and drawbacks to each service. For the sake of discussion, a public channel is considered to be any communication media not provately owned. The AUTOVON system in the military structure is considered a public channel since it is not wholly owned by the Department of Defense. AUTOVON is a means of transmission available in either a dial-up or dedicated mode of operation. The following discussion of each type of service applies to the AUTOVON system, with differences noted. Also, discussion is limited to AT&T standard dial-up service and voice grade lines.

(1) Dial-up Service. The standard dial-up service used in everyday voice communications can be used to transmit data. The line will not be removed from voice service and the standard instrument may remain as an integral part of the network. As a result, a connection could be made through any telephone. When service is desired, a call is placed to the host computer site. The handsets are then plugged into special modems at the terminal and computer sites (Figure 3-49). The terminal is then on-line to the computer.

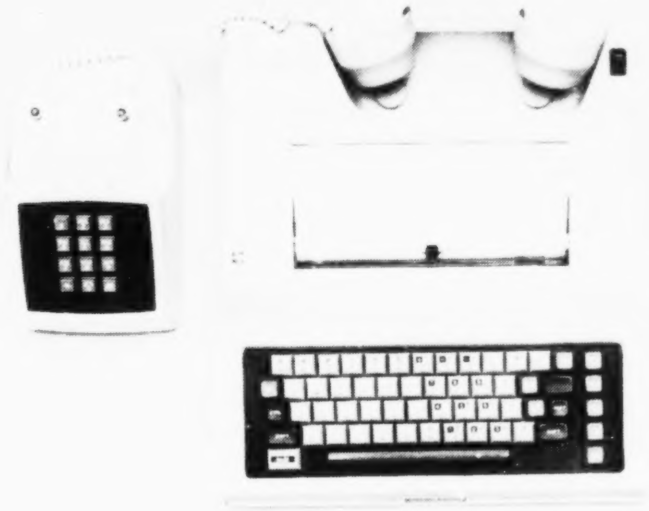


Figure 3-49. Portable Data Terminal
(Texas Instruments Model 745)

When the data transmission is completed, the connection is broken by simply hanging up the telephone. Characteristics of the dial-up mode are listed in following paragraphs.

(a) Automatic Routing. The actual path the call takes is established at the time the call is placed. The next time, the call may be routed a different way. If one circuit is busy, the call will be switched or routed along a circuit not in use. The switching is automatically done outside the user's control. In fact, the user will not have any idea how the call is routed.

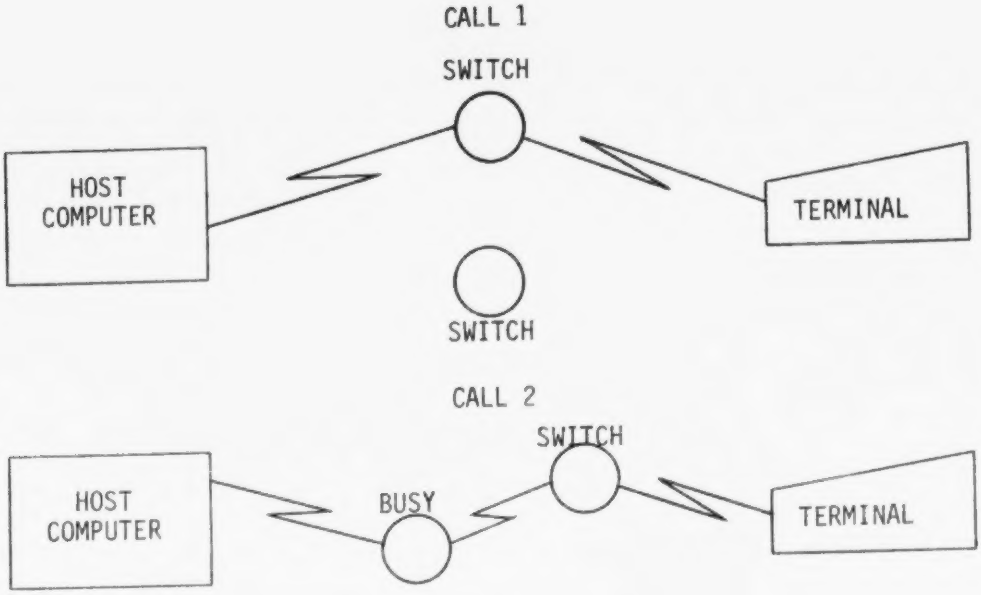


Figure 3-50. Automatic Routing.

(b) Backup With Other Lines. This is based on automatic routing. If something happens to the communications lines presently in use, the call can be routed over a working line simply by redialing. Using the dial-up mode, AT&T would have their entire telephone network available as backup.

(c) Cost. As in normal telephone service, the cost will be for the duration of the call, or, from the time the connection is made until it is broken. The charge is based on standard long distance rates. No increase is made because the voice grade line is being used for transmission of data. If an AUTOVON circuit is used, there is no cost incurred because the service is leased by the Department of Defense.

(d) Transfer Rate. Using the standard AT&T switch network, the maximum transfer rate is 4,800 bits per second. The limiting factor is more the capability of available modems than the capacity of communications lines.

(e) Half Duplex Transmission. The normal switching network is two-wire only, which restricts transmission to half-duplex. Provisions are available for full duplex (4-wire) by establishing two telephone calls simultaneously.

(f) Slow Turnaround. Because of the special equipment needed for voice communication, such as echo suppressors, the turnaround time for data transmission may be in excess of a few hundred milliseconds.

(2) Dedicated Service. Voice grade lines are also available in a dedicated mode. In this mode, the line is not supporting voice communication but is supporting data transmission. In this respect, the control signal and echo suppressor needed for voice communication will not be present, allowing for more efficient use of the channel. The characteristics of dedicated lines are as follows.

(a) Permanent Routing. A permanent connection will be made between the terminal and the host computer. This line will always be open, regardless of whether or not it is used. At the time of the contractual arrangement with the communication facility, the switches are hardwired to establish the route. The reason is because the fixed route is available only to the customer leasing that line. In a dial-up mode, however, if one customer is not using the line, it can be used by another customer.

(b) No Backup. If a circuit is hardwired, there is no way to hang up and redial to establish a new route. If the line is down, service is disrupted until the line is placed back into service. The only way to provide backup is to lease another line or have a dial-up as an alternate.

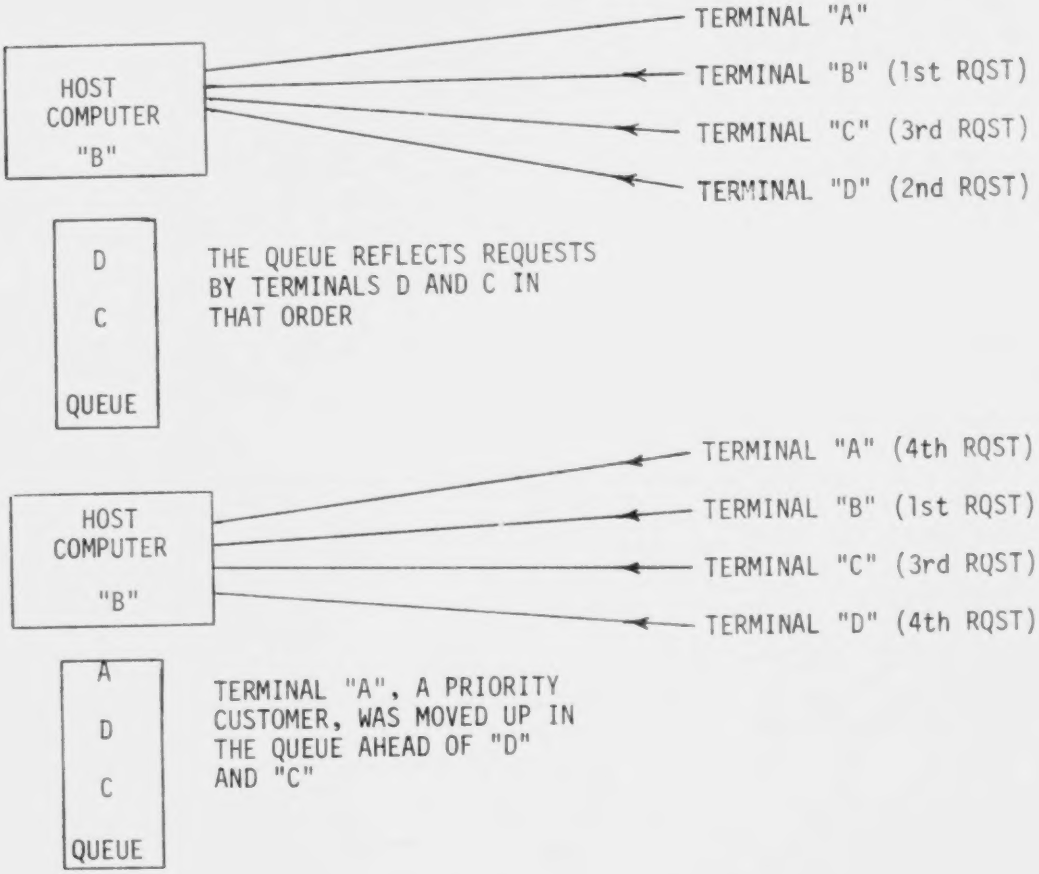
(c) Cost. The lines are leased; therefore, the user is continually paying for the service whether the line is being used or not. Charges are based on the type of line, and service per mile per month.

(d) Transfer Rate. Using the standard AT&T private line, the maximum transfer rate is 10,800 bits per second.

(e) Full Duplex Transmission. Full four-wire service is available allowing full duplex transmission.

3.46. ACCESS METHODS. The network contains a host computer to provide the support, the terminals being supported, and communications lines to tie the two components together. How then does the customer use this network to gain access to the host computer? Some scheme must be built into the system to function in the capacity of controlling the access so the computer knows when a terminal needs support. Basically, there are two methods; contention, and polling.

a. Contention. In a network using contention as the access method, the terminal initiates a request to transmit. Based on that request, one of two actions will be taken depending on whether or not the system is busy. If the system is available, a signal is sent to the terminal to transmit, and the system is then available to that terminal for service. If, on the other hand, the system is busy at the time the terminal initiates a request, the terminal is recognized as needing service but is given a busy signal. To keep service orderly to waiting customers, a queue of requests is maintained under control of the system. A queue is nothing more than a waiting line such as one might encounter at a checkout stand in a supermarket. When a request for service is received and the system is busy, that request goes into the queue. Normally, requests in the queue are serviced on a first-in first-out basis. In some situations, there are priorities among the users, and one in particular may be recognized as quickly as possible. This poses no problem because the queue can be altered for priority users. When this type customer requests service and the system is busy, his request can be moved to the head of the queue. In this way, priority users would be recognized next (Figure 3-51).


HOST COMPUTER "B"
TERMINAL "A"
TERMINAL "B" (1st RQST)
TERMINAL "C" (3rd RQST)
TERMINAL "D" (2nd RQST)
D
C
QUEUE
HOST COMPUTER "B"
TERMINAL "A" (4th RQST)
TERMINAL "B" (1st RQST)
TERMINAL "C" (3rd RQST)
TERMINAL "D" (4th RQST)
A
D
C
QUEUE

THE QUEUE REFLECTS REQUESTS BY TERMINALS D AND C IN THAT ORDER

TERMINAL "A", A PRIORITY CUSTOMER, WAS MOVED UP IN THE QUEUE AHEAD OF "D" AND "C"

Figure 3-51. Contention.

b. Polling. In a network using polling as the access technique, the terminal is asked if it has anything to transmit. Inquiry of terminals is done on a predetermined sequence and time frames under control of the system. A polling list which designates the sequence in which terminals are to be polled is an integral part of the system. Based on this polling list, and the set frequency, terminals are polled. When it is "time" for a terminal to be polled, the system is interrupted, and the proper terminal is questioned. Consider a system supporting 5 terminals, A through E. These terminals are to be polled in the sequence A, C, E, B, D, at one-second intervals. At the 1-second mark, terminal "A" would be polled. On the 2-second mark, terminal "C" would be polled, at the 3-second mark, terminal "E", and so on. Finally, at the 5-second mark, with the polling of terminal "D", the polling sequence would be completed (Figure 3-52).

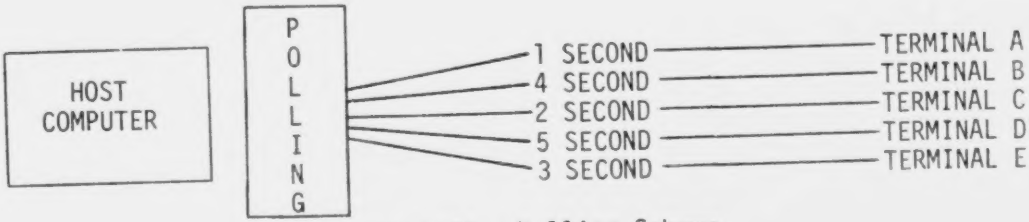


Figure 3-52. Polling Scheme.

As in contention, a queue may be maintained by the system to insure orderly service. If terminal "A" requires service on the 1-second mark, and the system is available, terminal "A" would be supported. If the system were busy, terminal "A" would go into the queue. If terminal "A" is using the system when the 2-second mark arrives, and interrupt occurs. Terminal "C" is now polled, and if it requires service, the fact is entered into the queue, as illustrated in figure 3-53. Priority among customers can also be taken into consideration under polling. If a high priority customer does exist, that terminal can be polled more often than any other of the terminals.

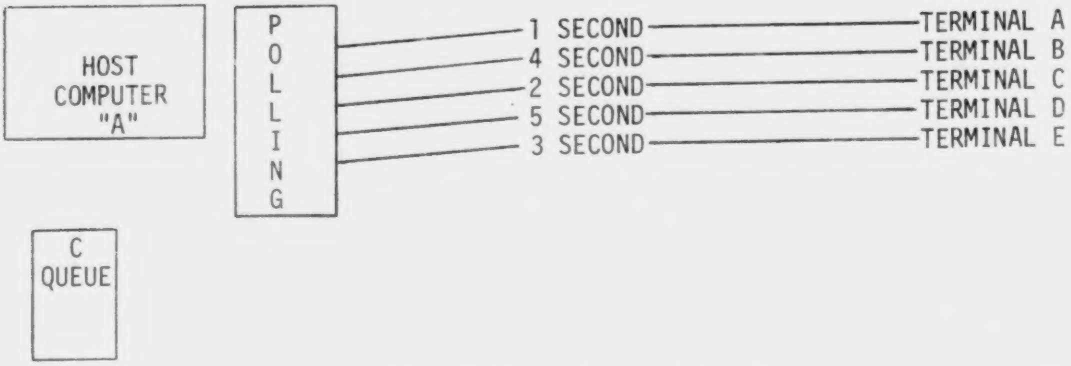

HOST COMPUTER "A"
POLLING
C QUEUE
1 SECOND
4 SECOND
2 SECOND
5 SECOND
3 SECOND
TERMINAL A
TERMINAL B
TERMINAL C
TERMINAL D
TERMINAL E

Figure 3-53. Polling Scheme With Queue.

Recall of the polling list is something that must be developed. It outlines for the system the sequence in which polling should take place. Using the example of five terminals again, terminal "A" has been designated a priority terminal. To provide terminal "A" with more service, the polling sequence could be changed so that terminal "A" would be polled every other time, as in Figure 3-54.

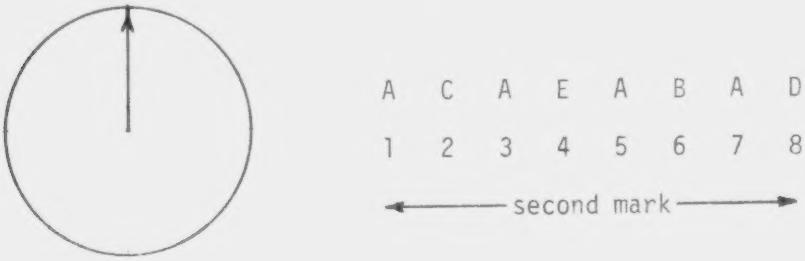

A C A E A B A D
1 2 3 4 5 6 7 8
second mark

Figure 3-54. Polling Scheme With Priority.

3.47. ERROR CORRECTION SCHEMES. Regardless of how foolproof a system might be, errors will still occur. Errors can be very costly to an organization, especially if those errors are allowed to infiltrate the master files. At times, decisions may be made based upon this erroneous information. Any goal the system must reach will be established by management, who must determine the degree of systems accuracy required. It is up to the systems analyst to design into the system the necessary hardware and software processes, coding structures, equipment, and procedures to accomplish the goal. When a standard system is functioning in a batch environment these kinds of problems exist, and they will still exist in an on-line system. But, because of the communications aspect, the problems will be greater. In the past, there were human errors and hardware failures, and they will still exist. Any time humans are involved in the preparation of data, errors will occur. Neither are the computer or the peripherals immune; they too can malfunction. The types of errors that existed in off-line systems may also occur in on-line systems. In on-line systems, an additional area where errors can occur is the area of transmission errors. This area rightly deserves much of the analyst's concern. Since there are no new techniques applying to human errors or hardware malfunctions, discussion of error detection and/or correction will be limited to transmission line errors.

a. Transmission Line Errors. Errors will occur, no matter how well a system is planned. These errors can be anything from noise on the line to malfunctions or atmospheric interference. Some types of signals are more susceptible to interruption or atmospheric interference than others. Microwave works on the line-of-sight principle. One antenna must see the other. If these antennae lose their alignment, errors will result. A severe electrical storm may occur somewhere along the network, changing a bit from a 0 to a 1. That's an error. According to AT&T, the chances of an error occuring averages 1 in 10^5 on a voice grade line transmitting at 4800 bits per second. At 9600 bits per second on a voice grade line the average error rate is 1 in 10^7. This level of error rate may not be significant, and will require no special effort to at least detect, and possibly correct, errors. In some cases, however, it will be significant and some of the errors will need to be recognized. If the system is to go so far as to correct the error or ask for retransmission will depend on the criticality of the information, and the user can accept a delay.

b. Methods of Detection and/or Correction. Since it is an accepted fact that errors are goingto occur, steps will have to be built into some systems to take care of this contingency. Various code structures are available to assist in the detection of errors, such as the 3-out-of-7 code. In this particular code structure, each character contains 7 bits and there will always be three bits on. If there are more or less than three bits on when a character is received, there is an error. Two methods of dealing with errors are listed below.

(1) Automatic Retransmission Request (ARQ). This method deals only with detection of errors at the receiving end and does not employ enough information to correct them. As the information is received, a check is made to see if errors occurred during transmission. A control signal is sent back to the transmitter notifying of either a good or a bad transmission. On notification of a bad transmission, the sending end retransmits the block that was in error. If transmission were half-duplex, the sender would wait for receipt of the control signal before sending the next block of data. If the signal were for a good transmission, the next new block would be sent; but if it were a signal for a bad transmission, the last block would be sent again. This method does result in a delay. If full duplex lines were used,

the data and control signals could be transmitted at the same time, eliminating the wait for the signal. Regardless of whether full duplex or half-duplex transmission is used, there must be something in the code structure that will allow identification of errors. One widely used means is a parity bit added to a character. This way, if odd parity is valid, but on receipt, a character had even parity, an error would be indicated. Some bit in that character had been altered in transmission. This type of error is illustrated in Figure 3-55. Using only the parity bit on the character does not provide enough information to determine which bit in the character has been changed. But now, some of the data being transmitted are non-information bits.

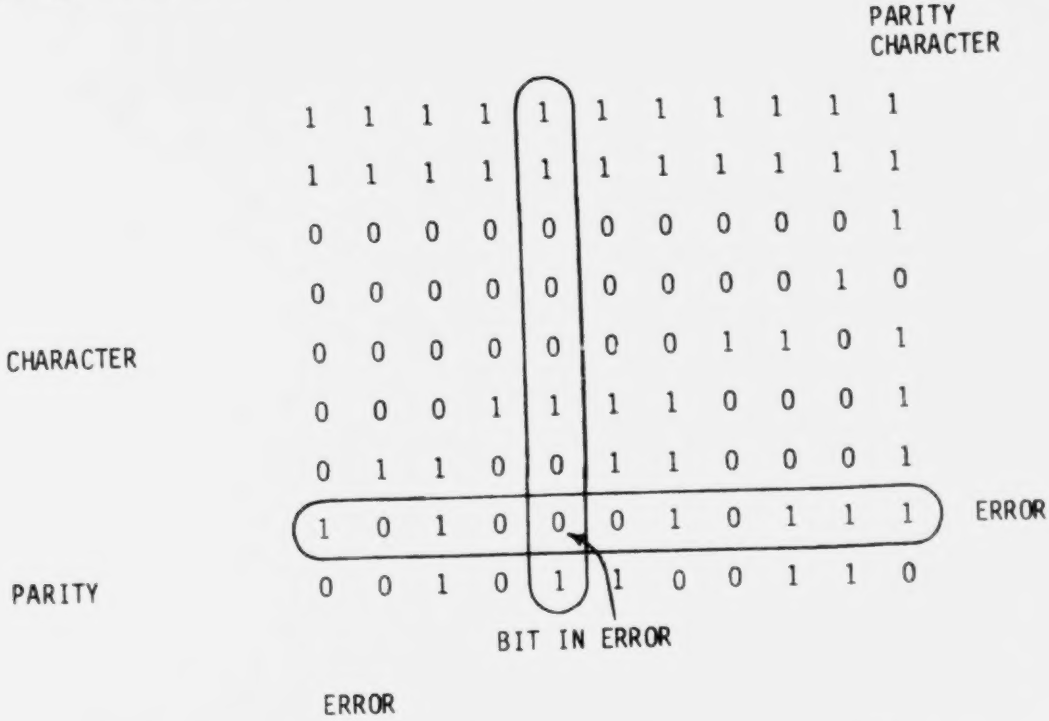
1 BLOCK

CHARACTER	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	1	1	0
	0	0	0	1	1	1	1	0	0	0
	0	1	1	0	0	1	1	0	0	0
	1	0	1	0	0	0	1	0	1	1
PARITY	0	0	1	0	1	1	0	0	1	1
					ERROR					

Figure 3-55 · Parity Bit - ARQ
(Data sent in EBCDIC)

(2) Forward Error Correction (FEC). This method not only employs additional bits to determine if an error occurred, but also determines the location of the bit in error. Since a bit has only two conditions (0 or 1), if the exact location can be determined then a bit can be altered or the polarization reversed. Of course, the deletion and correction is done at the receiving end, eliminating the need to go back to the sender. There will be no control signal to the sender indicating a good or bad transmission, but the number of non-information bits that will be transmitted is increased. For now there will be additional bits to identify which information bits are in error, as well as those needed to identify the location of the error. Again, parity can be used. To accomplish FEC, a parity

character is added at the end of the block. In ARQ, the parity bit in each character was used; with FEC, both the parity bit in each character and a parity character at the end of the block are required as shown in Figure 3-56. An error on a parity bit in the indicates that a bit within that character is in error. The error on a bit in the parity character indicates the row in which the bit is in error. As in any matrix, the intersection of the two indicators shows the actual bit in error. Now all that need be done is to correct the error by changing the state of the erroneous bit. In our example, it would be changed from 0 to 1. The only problem is in having enough information to locate and identify an error.



1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	1	0	1
0	0	0	1	1	1	1	0	0	0	1
0	1	1	0	0	1	1	0	0	0	1
1	0	1	0	0	0	1	0	1	1	1
0	0	1	0	1	1	0	0	1	1	0

Figure 3-56. Parity Bit and Character - FEC.

FEC requires many more non-information bits than ARQ. Care must be exercised when establishing the error detection method. For instance, if the parity character technique just discussed is used, it affects the number of characters in the block. When speaking of a number, reference is made to whether the number of characters in the block should be odd or even. The parity character is to be ignored in this count; only the information characters should be considered. The selected code structure is really the determining factor. Two rules apply for parity to work properly.

(a) If the number of information bits in a character are even, the number of information characters in a block should be even.

(b) If the number of information bits in a character are odd, the number of information characters in a block should be odd.

If these rules are not followed, the parity of the block will be invalid, and the block will have a built-in error at the time it is generated. Parity is based on the number of bits being odd or even. Both conditions cannot exist at the same time if the data is correct. Figure 3-57 shows a block with the proper number of information characters. The code requires an even number of bits (8) to represent a character. Following rule 1, the block contains an even number of information characters. Odd parity was selected for the example. No problems were encountered at the time the block was generated.

BLOCK SIZE

	INFO CHARACTERS										PARITY CHARACTER
INFO BITS	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	1	1	0	1
	0	0	0	1	1	1	1	0	0	0	1
	0	1	1	0	0	1	1	0	1	1	1
PARITY	0	0	1	0	1	1	0	0	1	1	0

Figure 3-57. Info Bits - Even / Info Characters - Even

Figure 3-58 is an example of block size not conforming to rules. The code structure for a character was not changed. What was changed was the number of information characters in the block, now it is odd. The question now is, "What condition is the parity bit in the parity character?" This rule violation gives a built-in error, always resulting in a bad transmission.

	INFO CHARACTERS									PARITY CHARACTER
INFO BITS	1	1	1	1	1	1	1	1	1	0
	1	1	1	1	1	1	1	1	1	0
	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	1	1	1
	0	0	0	1	1	1	1	0	0	1
	0	1	1	0	0	1	1	0	0	1
	1	0	1	0	1	0	1	0	1	0
PARITY	0	0	1	0	1	1	0	0	1	?

Figure 3-58. Info Bits - Even / Info Characters - Odd.

3.48. NET DATA THROUGHPUT. Management may be misled into believing that information will be transmitted at one speed when in reality in will be transmitted at some different one. Thre are a good many factors which affect the throughput, especially if ARQ or FEC are being supported. More than the user's information will be transmitted, errors will occur, there will be a time lapse between blocks, etc. These factors and others must be taken into consideration, for the systems analyst is interested in, and must develop the net data throughput for the system. Everyone may think that the network being designed will be carrying 4800 bits per second in information but it is going to be somewhat less than that. If it is necessary that 4800 bits per second be obtained, then adjustments will have to be made in the network. The formula for calculating net data throughput is:

$$NDT = M\,(1-K)\,(1-P)\ /\ ((M/R)+T)$$

Where:

M = Transmitted Block Length Expressed in Bits
R = Channel/MODEM Normal Operating Speed Expressed in Bits per Second
P = Block Retransmission Probability (Block Size X Rate of Error)
T = Time Between Blocks Expressed in Seconds
K = Fraction of Non-Information Bits

M = 10,000 bits
R = 4800 bits per second
P = .1
T = .2 seconds
K = .05

Thus, NDT = 10,000 (1-.05) (1-.1) / ((10,000/4800)+.2)
= 10,000 (.95) (.9) / ((2.0833)-.2)
= 10,000 (.855) / 2.2833
= 8550 / 2.2833
NDT = 3744.58 bits per second

This is somewhat less than the 4800 bits per second that management may have thought the system would handle. It would appear that the easiest way to increase the throughput would be to increase the block size. However, increasing the block size also affects the value of P (retransmission probability). For instance, doubling the block size may have an adverse effect on the net data throughput. If M were changed to 20,000 bits, then -

$$P = \frac{20,000}{1} \times \frac{1}{100,000} = .2$$

and: NDT = 20,000 (1-.05) (1-.2) / ((20,000/4800) + .2)
= 20,000 (.95) (.8) / (4.1666 + .2)
= 20,000 (.76) / 4.3666
= 15,200 / 4.3666
= 3480.96 bits per second

Doubling the block size decreased the net data throughput. But what would happen if the block size were halved? If M were reduced to 5,000 bits, then -

$$P = \frac{5,000}{1} \times \frac{1}{100,000} = .05$$

and - NDT = 5,000 (1-.05) (1-.05) / ((5,000/4800) + .2)
= 5,000 (.95) (.95) / (1.0416 + .2)
= 5,000 (.9025) / 1.2416
= 4512.5 / 1.2416
= 3634.42 bits per second

In this case, 10,000 bits might be the optimum block length. Only one factor was changed in this example - the value of M. The point is that several alternatives must be examined quantitatively in great detail.

3.49. MODEMS. Discussion up to this point has led to the equipment within the network. It has been point out that a network will contain certain basic types of equipment for specific purposes. One such piese of hardware in the network is the MODEM. It has been mentioned sevewral times throughout this section and is an important part of the system. It is the interface between the (host computer/terminal) and the communications line. In very instances will on-line systems not include modems, or a piece of hardware with another name which performs the functions of the modem. The term modem is a contraction of two words, modulation and demodulation. There are two basic types of modems and cost will vary depending on the speed of transmission and the selected optional features. Regardless of type, each will accomplish the same primary functions.

a. Functions. The basic modem, regardless of type, will accomplish the following tasks.

(1) Digtal-to-Analog/Analog-to-Digital Conversion. The second step in preparing data for transmission is the generation of the analog signal or sine wave. This then becomes the carrier for the data, allowing it to be transmitted via the telephone system. At the receiving end, the carrier must be converted back to the digital format. This function is accomplished in conjuction with the first function. The modem, starting at the sending end, accepts digital data from the hardware, generates the analog signal, and releases the carrier to the communications line. Upon receint, the modem will generate the digital format. It is the analog signal which carries the data from modem to modem. There are three different ways to modulate the carrier to represent the conditions 0 and 1: amplitude modulation, frequency modulation, and phase modulation. These are merly three different ways in which 0 and 1 conditions can be represented using a sine wave.

(a) With amplitude modulation, a zero can be distinguished from one by verying the ampitude. A one will have a high amplitude, while a zero will have a low amplitude.

(b) In frequency modulation, the frequency of the carrier is altered so that one frequency represents a one, and a different frequency represents a zero.

(c) Phase modulation uses the technique of reversing the phase. The values zero and one are represented by the carriers being 180 degrees out of phase with each other.

(3) Line Equalization. It is a fact of life that a wide variance in the grade of service available through the public telephone system exists. Distortion of the signal may occur, and there are two ways to overcome this distortion. If the line is leased, the telephone company can condition the line. In the case of dial/up, line conditioning is not possible, and other methods will be required to accomplish the same thing. This can be done at the modems. On some modems, this will be a manual operation, and the modem will require fine-tuning every few months. On other types of modems, particularly those capable of transmitting more than 4800 bits per second, the operation will be automatic. The reason for this is that the greater the speed, the greater the need for equalization of a modem.

(4) Control Strength of Signal. Over a given distance, the signal will begin to fade or lose its strength. When the signal is received it may need to be increased so that it can be interpreted properly.

b. Types of Modems. A variety of manufacturers market modems, but a modem can be classified by the type of interface used between it and the communications line. From this perspective, there are three types of modems.

(1) Acoustic Couplers. The acoustic Coupler (Figure 3-59) is used primarily in those cases where a portable terminal is desired. In some cases the acoustic coupler has been integrated into a portable terminal. In all cases, the primary connection between the terminal and the communications line is the standard telephone instrument. To go on-line, the user dials the host computer. When the connection has been established, the handset is plugged into the acoustic coupler. At the conclusion of the transmission the handset is removed from the coupler and hung up, thus breaking the connection. Using this type of modem, a user would have access to the computer from any place there is a telephone. The acoustic coupler is normally limited to low speed devices operating in an asynchronous mode.

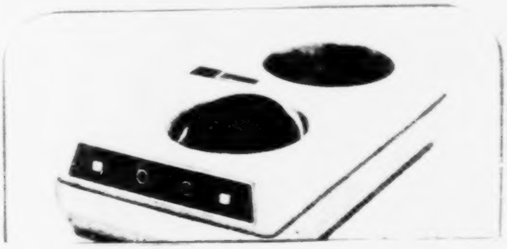


Figure 3-59. Acoustical Coupler.

(2) Data Set. Another type of modem which uses the dial-up capability as did the acoustic coupler is the data set (Figure 3-60). The similarities do not end at that point. A data set is capable of supporting high-speed synchronous data transmission, whereas the acoustic coupler is primarily for low-speed asynchronous transmission. The size of the data set does not lend itself readily to use in a portable situation, but is more for use in a fixed installation where dial-up service is necessary. Associated with the Data Set is the Data-Phone (3-61), which comes with either a rotary dial or touch-tone. Dialing the host computer site through the data phone establishes the communications link. Once the connection is made, the user pushes a button on the phone marked "DATA". When the data-phone is in the data mode it is capable of sending or receiving data.

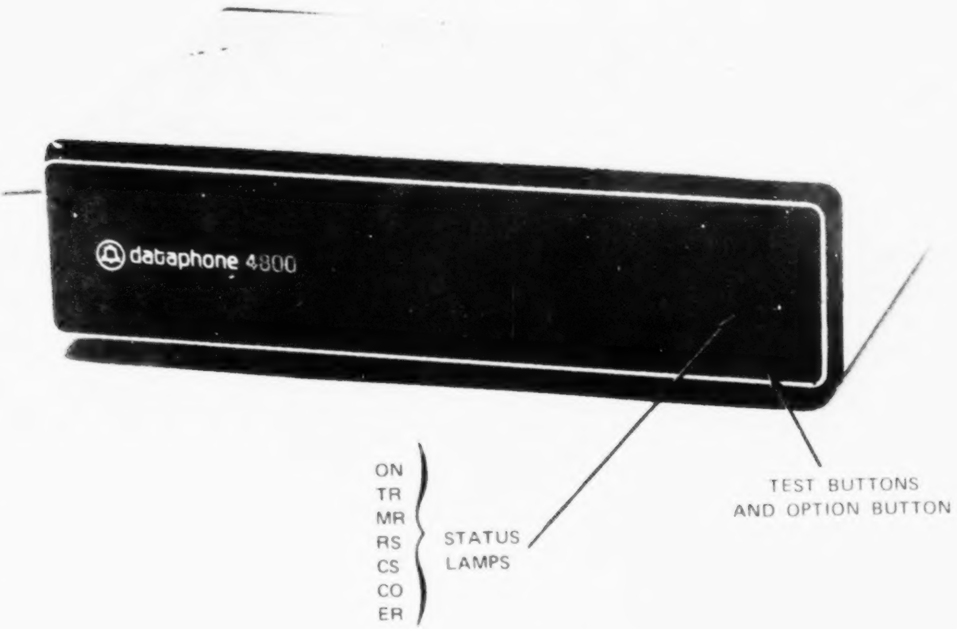


Figure 3-60. Bell System Data Set 208B.

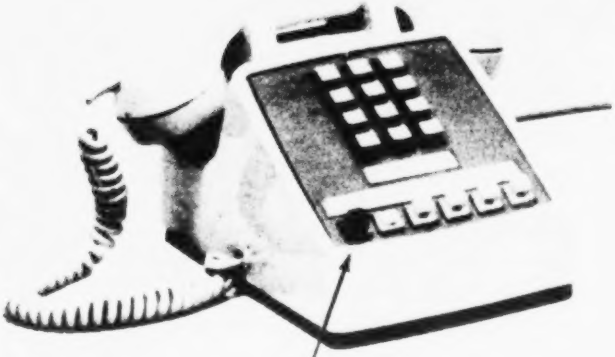


Figure 3-61 . Data-Phone.

(3) Hardwired Modem. The last type of modem is used in those networks employing dedicated lines. With the hardwired modem (Figure 3-62) a permanent connection is made with the communications line, eliminating the telephone instrument. In this type operation, there is no dialing to establish the connection. The terminal is on-line, being serviced via a dedicated line.

ICC
MODEM 4700/48

Figure 3-62. Hardwired Modem.

c. Cost. The cost of a modem depends on what is it capable of performing and the speed. The acoustic coupler is the least expensive means of interfacing. At the low end of the scale, a modem capable of handling 300 bits per second will cost approximately $200 for a basic unit. For modems capable of handling data at speeds of over 2400 bits per second, the cost will be in the neighborhood of $1 per bit per second. More than just the raw cost of the modem should be considered. A comparison should be made between cost, speed, and error control. Speed and error control have a definite impact on cost. The increased cost may be offset through greater efficiency of the system.

d. Optional Features. In addition to the basic functions, with optional features, a modem can perform additional functions. Like any money items, optional features will add to the cost of the modem. Some of the available options are listed below:

(1) Automatic Answer/Dial-up. This option allows unattended operation. With this feature, incoming calls to the terminal would automatically be answered if the terminal is not already busy.

(2) Automatic Diagnostics. This feature will allow the monitoring of the modem's circuit. The diagnostic test can be expanded to include the line and the modem at the other end.

3.50. TELECOMMUNICATIONS CONTROL UNIT (TCU). Another piece of hardware in the network is the telecommunications control unit (TCU). Sometimes referred to as the terminal control unit or line control unit, this piece of equipment is located between the host computer and the modem, and acts as the interface between the two, just as the control unit in the computer interfaces between the CPU and the I/O devices. The TCU is the same type of device. It is the control unit, or hardware interface, between the CPU (host computer) and the terminal. A TCU may or may not be required at the terminal end. For example, if the terminal is nothing more than a portable keyboard with hard copy capability, a TCU will not be used. If, however, the terminal was programmable on another computer, a TCU will be located at that end. Regardless of the location, host computer end or terminal end, the TCU is designed to perform specific functions. It is hardwired to perform these functions and is difficult to alter. The functions or duties of the TCU are as follows.

a. Code Conversion. The TCU handles several terminals, each of which may require a different code structure, while the host computer requires a different code from the terminals. The code conversion could be done in the host computer, but why tie it up? The TCU will perform code conversion under the control of the host computer. The reverse may also apply. The host computer generates information in one code, but the terminals need it in different codes. Again, the code conversion would be performed by the TCU.

b. Error Detection. The TCU can be hardwired to perform error detection or to recognize parity errors. That's the limit, because the TCU cannot perform error correction or data editing. The ARQ method of error detection may be supported by a TCU, but FEC would require action on the part of the host computer.

c. Access Method. Polling of terminals is done by the TCU under control of the host computer. The host computer initiates the operation and it is continued by the TCU. This relates to an I/O function. A read instruction is issued by the CPU, but it is actually accomplished by the control unit between the CPU and the I/O device.

Although the TCU performs these functions and relieves some of the administrative tasks, a great deal is still left up to the host computer. The functions that are still under control of the host computer go so far in some cases that the host computer must initiate the action, such as in polling. In addition, there are other disadvantages. There are limits to the number of lines which can be supported. This limit is really imposed by the host computer. For each communications line, a multiple or subchannel from the host computer to the TCU is required. It is a one-for-one type situation.

3.51. SOFTWARE. The last component of the basic network is the additional software required to support a teleprocessing environment. A standard operating system such as DOS or OS is not capable of supporting a teleprocessing system without some type of additional software - a teleprocessing control program (TCP). It should be apparent by now that this type of system does entail a large and complex hardware and software system. Supporting the hardware is the teleprocessing control program which acts as the interface between the hardware and the applications programs (Figure 3-63). An operating system is still required, and it will be responsible for exercising the necessary control over the entire system. Supporting a teleprocessing application requires an expansion of the system. Normally, the existing operating system will require modifications, with enhancements to support teleprocessing. Additionally, the system must be able to support and to operate in a multi-programming environment. This is necessary because the teleprocessing control program will occupy a partition and be on-line at all times when this function must be supported. How much memory the control program will require depends on the tasks which must be supported by it.

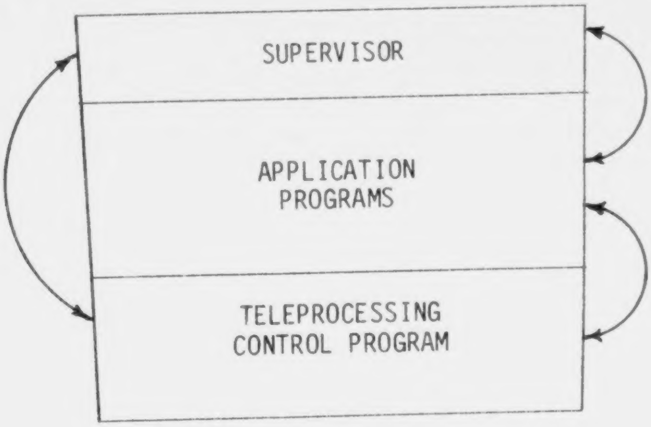


Figure 3-63.

A task may not apply to a system, may be supported by the control program, and actually be accomplished elsewhere, or may be accomplished by the control program. Some of these tasks are discussed below.

a. Terminal Recognition. The access method may actually be accomplished by the TCU, but it will be controlled by the software. The software will maintain the polling list and be responsible for initiating the action in the proper sequence. If contention is used as the access method, the software must be capable of recognizing when a terminal needs service.

b. Maintaining The Queue. some order has to be maintained to insure that the terminals receive equitable service. The maintenance of the queue, moving messages in and out, and making adjustment for priorities, falls to the control program.

c. Message-to-Block/Block-to-Message Conversion. The network looks for a block. Data is transmitted by block, either one or more characters, depending on whether the transmission is asynchronous or synchronous. The message to be transmitted will, more than likely, be of a different length than the transmission block. If the message is longer, then it must be broken into block-sized segments. On receipt, the blocks must be reassembled into the message format. The block length is normally a fixed element for a system, while messages can vary in length depending on the job. The system must be capable of recognizing when the end of a message occurs when receiving data, and of generating the end-of-message signal when transmitting data.

d. Error Detection or Error Detection and Correction. Although error detection can be performed by the TCU, the responsibility for, and control of, error detection lies with the control program. Normally, the control program will initiate the signal to the sending end indicating a good or bad transmission. If the host computer is the sending end, the control program is responsible for initiating retransmission when notified of a bad transmission. The control program is quite involved with ARQ, and even more so with FEC. The TCU is not capable of correcting transmission errors. If this task is to be supported, it must be accomplished is the control program.

e. Terminal/Customer Validation. Security is perhaps one of the most difficult areas in an on-line system. Control must be exercised over the terminals and the users. In a dial-up system, any terminal could penetrate the network by having the proper telephone number if there are no other identification procedures. Depending on whether classified or proprietary information is maintained, there may be several levels of validation. The first level would be positive identification of the terminal. Only those terminals authorized service should be recognized and granted access to the system. The next level has to do with the information being stored in the files. Everyone having access to the system does not necessarily have authority to access all the files. No longer does the situation exist where there is a human being acting as the go-between, asking the questions and determining if the user is valid and has authority to access a given file. This responsibility now belongs to the control program. It must determine if the terminal is valid; if the user is valid, and if the user is authorized access to desired information. Information being stored in the system must be protected. A user has entrusted the system with his information. If the system is to survive, that trust cannot be violated by allowing unauthorized access or manipulation. The responsibility of granting access, determining degree of access, and stating whether a user can read and write, or only read, a file rests with the owner of the information. An owner might grant read-only permission for access to his files, but not allow anyone to write to them. The fact remains that whatever the degree of validation, the control program must be capable of applying and enforcing the restrictions.

3.52. METHODS OF OPTIMIZING. Telecommunications systems can be very costly. All avenues must be examined during the design phase to establish the best network for the least amount of money. One of the most expensive portions of the system can be the communications lines. For example, over a distance of 1,000 miles a line capable of half-duplex transmission could cost approximately $700 per month. If the system were expanded to support 100 terminals each with a separate communications line, the cost for the lines alone would be $70,000 a month. Then, there would be the cost of modems; a modem capable of supportiong a low-grade line would cost about $175. For a 100-terminal system, 200 modems would be needed at a cost of $35,000. Techniques are available, however, which might reduce the number of lines and modems. The techniques selected will depend a great deal on such factors as volume of data, physical location of customers, desired speed of transmission, etc., and may vary anywhere from reorganization of communications lines to the use of new types of equipment such as multiplexors, front-end processors, data concentrators, and the like.

a. Multidrop/Multiport. These techniques are used in those situations when volume tends to be low. Using a network where the maximum customer usage was found to be 75 bits per second with an average of about 10 minutes per customer, it is seen that no one user will tie up the network. This might be a situation where either the multidrop or multiport technique can be used. What needs to be examined is the physical location of the users. Look for those cases where the locations seem to form a cluster or to follow a path. By employing a party line concept, the number of communications lines, and possibly the number of modems, can be reduced. The major drawback to the multidrop or multiport network is the same as any party line: if the line is being used by someone, everyone else on the same line must wait for it to be free before trying to use it. The same situation exists in teleprocessing using multidrop-multiport systems - if one terminal in the network is using the line, the remaining terminals must wait. In cases where system availability is critical, such as a transaction-oriented or retrieval system, these techniques may not be usable. On the other hand, a batch input or output system could easily adapt them. Each terminal would have a specific time to be on the air, thus eliminating the interference with each other. In a network where the users are along a path, a single communications line may be used for the long line to the most distant terminal, with short lines dropping off to each of the terminals. This is multidropping (Figure 3-64).

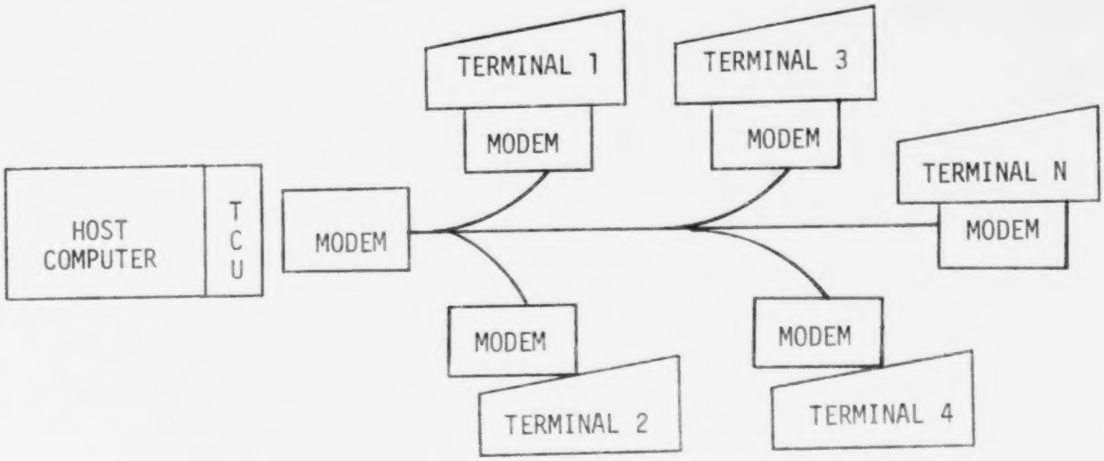


Figure 3-64.

Using this technique, the number of long lines and modems can be reduced, but now the users will be sharing a communication line to the host computer. The capabilities of the modem at the host computer end and the communication line will be based on the maximum speed required by a single user. In other words, if terminal 2 required a speed of 150 bps and the other terminals only required 75 bps, the design goal would be 150 bps. Another technique, multiporting, employs a similar party line concept, but results in changing the modems at both the terminal end and the host computer. This technique can be used when the terminals form a cluster, rather than following a path. With the customers in a cluster, the group can be serviced through one line with one set of multiport modems (Figure 3-65). A multiport modem is nothing more than a special with multiple ports (exit or entry points) on one side, and a single port on the other. The interface with the communication line is a single port. On the terminal side of the modem, each terminal has its own entry port. At the host computer end, the terminals have corresponding port assignments. As far as the host computer knows, each terminal has a separate communication line; but, through the multiport modem, a single line will be shared. The problem still exists that when one terminal is using the line, the remaining terminals must wait.

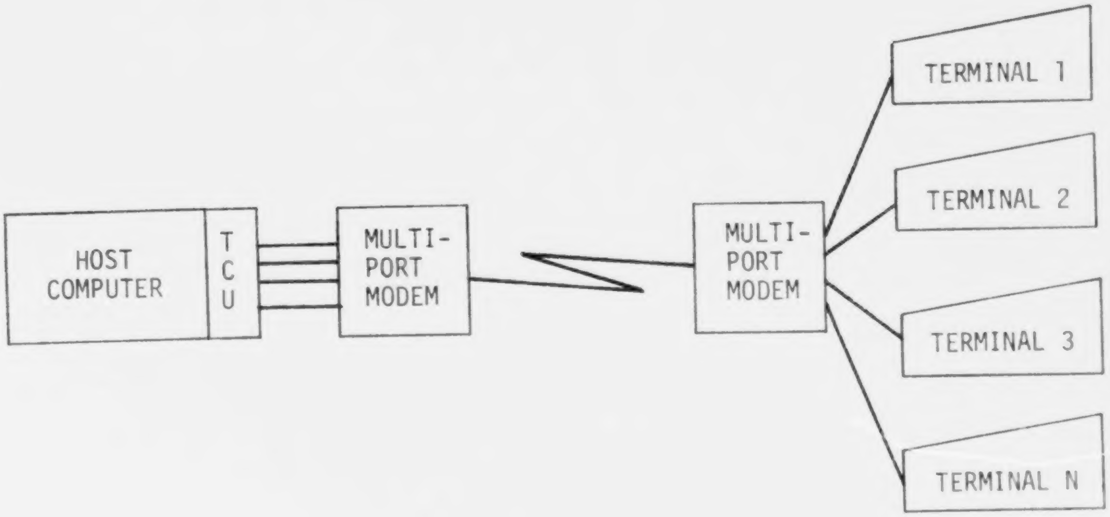


Figure 3-65. Multiport Network

b. Multiplexing. Turning to situations where volume is high, a different technique can be used to reduce the number of communications lines, yet cause no degradation in service to the user. It is important that the user not be sacrificed to save dollars. If, as in the previous example, usage were 18 to 20 hours per customer, then sharing a line, through multiporting or multidropping would not be sufficient. One solution might be the use of a higher speed line shared by the users. Refer again to the 100-terminal situation, where data was transmitted at 75 bits per second. Examining the price structure, a saving might be recognized if a higher line speed were used. There is a technique available that consolidates transmissions over a single line in such a manner that users will notice no difference in the service being provided. This technique, called multiplexing, takes advantage of higher line speeds to support several low speed terminals. Some hardware changes must be made to the network, together with a reduction in the actual number of communications lines, to support multiplexing. Special devices, called multiplexors, are used, which allow a single line to support several terminals. When multiplexors are used, there must be one at each end of the long line. Multiplexing is the division of a communications line into more than one channel. There are two techniques of multiplexing; frequency division multiplexing (FDM), and time division multiplexing (TDM). The two techniques are not competitive with each other because each falls into a different category of usage based on parameters or needs established during the system study. It is not uncommon to find both techniques used in the same network. Each has its place based primarily on three considerations: channel capacity, equipment costs, and line configuration.

(1) Frequency Division Multiplexing (FDM). FDM applies the concept of taking one channel and dividing it into several narrower channels or bands. Imagine a large pipe capable of carrying 3000 units. This pipe encloses several smaller pipes, each of which can carry 100 units (Figure 3-66). Another example of frequency division multiplexing is represented by the AM or FM radio dial. The AM dial covers a certain band width which is divided into several frequencies or bands. Each station is assigned one of the available frequencies for broadcasting. In FDM, the channels or bands are called derived channels or data bands. The capacity, or data rate, of each derived channel depends on the width of the derived channel. A relationship exists between the transfer rate and the width of the channel. The narrower the channel, the fewer the number of bits per second. In this respect, a derived channel of 75 bits per second will be narrower than a derived channel capable of carrying 134 bits per second. Dividing a channel in this way provides separate paths to the channel. Users would notice no difference in service from a separate line when serviced by this method. Since each terminal has, in effect, its own separate path, frequency division is said to be code-transparent. That is, each data band could use a different code because there is no interference between bands. Each terminal could use a code structure based only on that terminal's requirements, disregarding what the requirements might be for other terminals. The characteristics of volume and speed make frequency division multiplexing appropriate for low-speed asynchronous terminals, and FDM can be used in situations where the terminals either form clusters or follow a path. In the basic system applying FDM, the multiplexor takes the place of the modems and performs all the functions of the modems as well as providing derived channels (Figure 3-67). Now, one communications line would satisfy the communication needs, thereby reducing the costs for lines and modems.

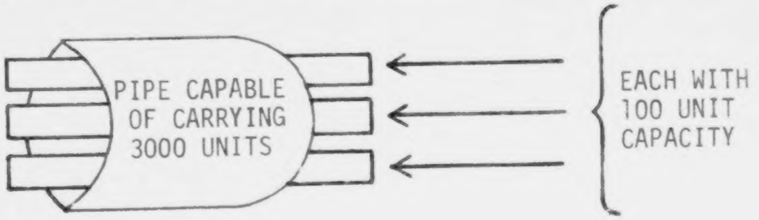


Figure 3-66. Frequency Division Multiplexing.

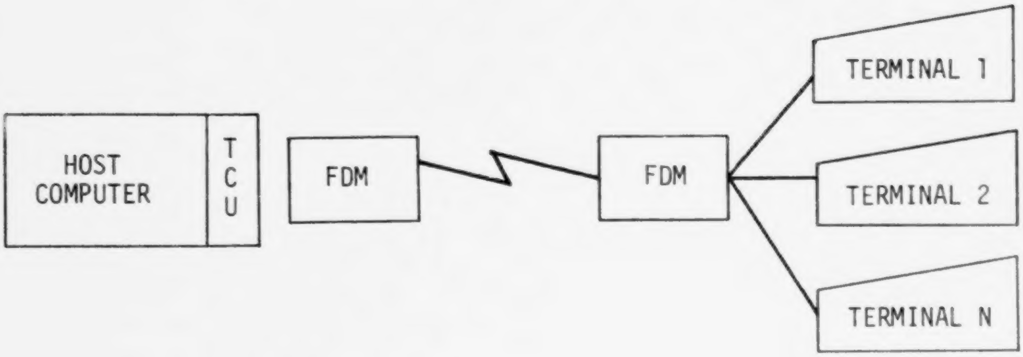


Figure 3-67.

(2) Time Division Multiplexing (TDM). In time division multiplexing, instead of the channel being divided into smaller bands or derived channels, it is divided into time segments. Each terminal has a segment of the total available time. The information from the terminals is accumulated, one character at a time, generating a continuous, synchronous bit stream (Figure 3-68) ready for transmission over a high-speed line. At the other end, the stream is broken apart and the messages reassembled. In this way, several terminals can be serviced by one communications line with no degradation in service, just as in frequency division multiplexing. There are other differences between the two techniques. In frequency division multiplexing, the modems were eliminated because their function was performed by the multiplexor. In time division multiplexing, that is not the case. High speed modems are required as an interface between the TDM and the communications line (Figure 3-69). The limiting factor regarding maximum transmission speed depends on the capability of the modem, not that of the multiplexor. Using time division multiplexing, it would be possible

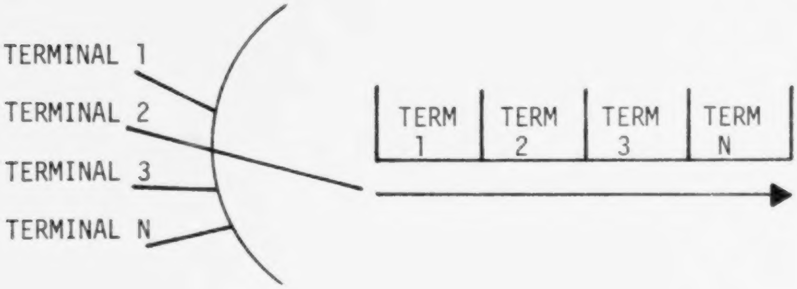


Figure 3-68.

to achieve 9600 bits per second on a voice grade line with proper conditioning. In this respect, greater speeds can be achieved with time division multiplexing than with frequency division multiplexing.

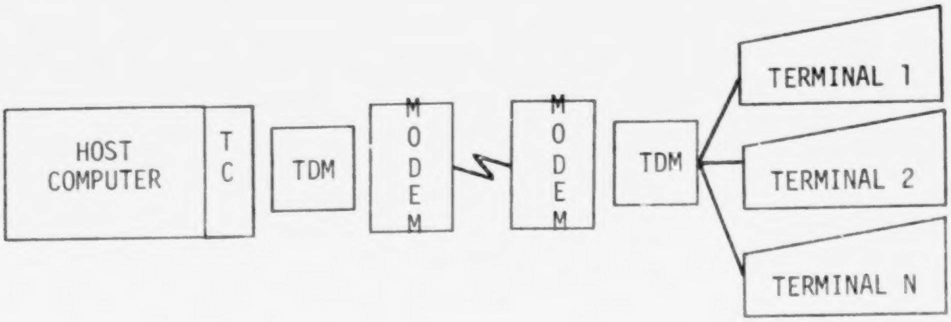


Figure 3-69. Network with Time Division Multiplexing.

c. Remote Data Concentration. Situations may exist where special steps must be taken to control clusters of terminals. High error rates may exist. There may be a requirement that terminals within the cluster communicate with each other. It could be that the expected peak volume requires some smoothing technique. The role of a remote data concentrator (RDC) is to take care of these problems. Remote data concentrators are minicomputers and can be used in place of, or in addition to, modems or multiplexors at the terminal end of the network (Figure 3-70). Since the RDC is programmable, many jobs can be accomplished at the RDC, rather than transmitting the data all the way to the host computer and accomplishing the necessary tasks at that location. Editing of data coming from the terminals can be done right at the RDC, catching errors near the start of the system, and returning the erroneous data to the terminals for correction. Only data which passes the edits would be transmitted over the line to the host computer. An RDC could be used as a collecting point for the information as

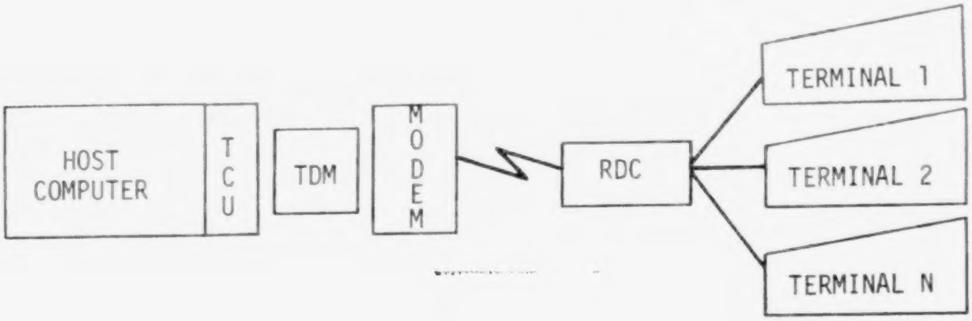


Figure 3-70. Network with a Remote Data Concentrator.

it is being generated at the terminals, with the terminals being nothing more than asynchronous key input stations. Recalling what has been said previously regarding synchronous and asynchronous transmission, each has a place in the transmission of data. To maximize the utilization of any long voice-grade lines, synchronous transmission should be used. Many terminals used will be the start/stop type not allowing synchronous transmission. By routing the data into a collecting point via a short line to a concentrator, only the short lines would have to support asynchronous transmission. From the host computer to the concentrator synchronous transmission over high-speed voice-grade lines would take place. If a terminal needed to send information to another terminal using FDM's the message would have to go all the way to the host computer, where it would be routed under control of the computer to the appropriate recieving terminal. If an RDC were used in place of these, the RDC could perform the message routing.

d. Programmable Front End Processor (PFEP). Another use of the minicomputer has been able to replace the TCU. Using the minicomputer as a programmable front end processor in lieu of the hardwired TCU results in benefits from cost savings and in increased responsibilities. A TCU is hardwired to perform specific tasks; but a PFEP can be programmed to perform various tasks from editing of incoming data, error detection and correction, message routing, etc. In this way, a good deal of the processing overhead is shifted from the host computer to the PFEP. Also, a savings will be noticed in the number of channel addresses. Only one channel address is needed between the PFEP and the host computer, while many are required between the host computer and a TCU. This frees channel addresses for other I/O peripherals. All of this together results in a real savings in both memory and execution time where the host computer is concerned.

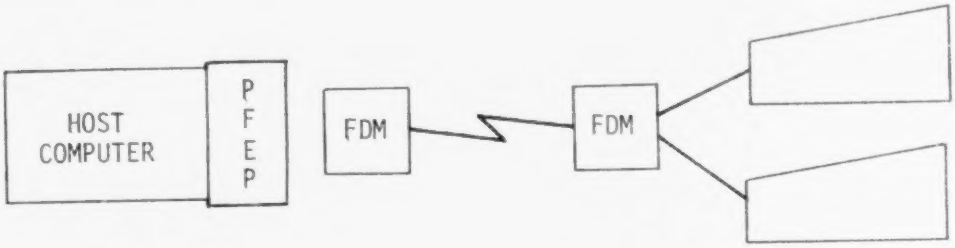


Figure 3-71. Network with Programmable Front-End Processor.

3.53. SUMMARY. The unimaginable speeds at which vast amounts of data may be processed are useless unless the user receives his information in time for it to influence his decisions. For this reason, teleprocessing techniques are a vital part of modern data processing systems. The systems are characterized by medium to large-scale computers, special and complex hardware, complex software, and a high degree of systems reliability. In the future, more and more systems will employ terminals that allow the user to communicate directly with a computer located elsewhere. In this way, teleprocessing techniques can be the answer to the question of how to get the information to where it is needed at the time it is required to be there.

CHAPTER 4. SYSTEMS DOCUMENTATION

SECTION 1. GENERAL

4.1. REFERENCES:

a. TB 18-111.

b. TB 18-103.

c. DOD Standard 7935.

4.2. PURPOSE. This Student Handout will give you a ready reference for documentation standards, forms, and methods which can be used to assist you in the production of systems documentation in accordance with current Department of Defense and Department of the Army standards.

4.3. SCOPE. Information presented here is applicable to systems documentation produced in support of DOD-designated ADS, Standard Army Multi-command Management Information Systems (STAMMIS), and command-unique systems.

4.4. IMPLEMENTATION. The documentation standards presented in DOD Standard 7935 (Automated Data Systems Documentation Standards), and TB 18-111 (Army Automation Technical Documentation) should be implemented by January 1980, unless a waiver ha been requested from US Army Computer Systems Command. Systems which will be documented under DOD 7935 are:

a. All new or previously undocumented ADS.

b. Systems which are currently documented and are undergoing a major revision. A major revision exists when 40% or more of the pages of a documentation manual are affected by new or acumulated changes.

c. STAMMIS and DOD-designated ADS.

d. Systems which have been documented to standards other than DOD 7935, and which are not undergoing major revision do not require redocumentation. These systems must be reported through command channels to USACSC.

e. Compliance schedule. A request for waiver must be submitted to HQ, USACSC if the following schedule cannot be met.

(1) New ADS will be documented in accordance with DOD Standard 7935 as they are developed, and will be completed within six months after the start of the deployment phase (as used in TB 18-100).

(2) For previously undocumented ADS, or for systems undergoing a major revision, documentation according to the DOD standard will be completed within six month of the publication date of TB 18-111 or the start of the deployment phase (whichever is later).

4.5. OBJECTIVES. The objectives of the DOD Standard is to provide guidelines for the development and revision of systems documentation, and to provide the descriptions of each of the technical documents to be produced during the life cycle of an ADS.

4.6. SIZE OF DOCUMENT TYPES. The document types prepared for any given project may range from only a few pages to several hundred pages in length. The size of any documentation package depends on the magnitude and complexity of the project; the opinion of the project manager as to the level of detail he feels is necessary in the documentation; the environment in which the system will operate; and other factors. Each of the Sections within a particular document type will, however, generally be related to the other Sections of the same document in the percentage of the overall volume that they make up. While these percentages may vary, the chart on the following page provides an approximate relationship of the volume of each Section of a particular type of document to the other sections within the same document.

4.7. DOCUMENTATION DEVELOPMENT POLICY.

a. Documents required. While the intent of DOD 7935 and TB 18-111 is to develop only the documentation needed to support an ADS throughout its life, Department of the Army policy requires that all 10 document types identified in DOD 7935 be developed as individual manuals unless supported by an approved waiver. Waiver requests to produce fewer than the 10 document types may be granted by major commands when the waiver request is accompanied by the complexity chart described in section 3 of this chapter.

b. Document conversion. Existing ADS documentation may be converted to conform with DA and DOD standards on a phased basis. The primary emphasis should be to replace existing documentation with a Users Manual and a Computer Operations Manual prepared in accordance with TB 18-111. Other existing documents may be phased out as manuals prepared to TB 18-111 standards are completed.

c. Section and paragraph titles. The section and paragraph titles and numbering scheme depicted in the table of contents for each document type in DOD 7935 will be adhered to without resequencing, except as explained below:

(1) When the information specified in a 2-position numbered paragraph is not applicable, insert "NA" after the paragraph number and title. Subordinate paragraph numbers and titles will not be documented.

(2) When none of the information in a particular section of a document is applicable, insert a brief reason for not documenting that section after the section number and title, and eliminate the paragraphs for the undocumented section.

d. Publication. All of the DOD 7935 document types may not require publication, however, when they are developed, they must be prepared in the format required by DOD 7935 and TB 18-111, and be suitable for reproduction.

SECTION NUMBER \ DOCUMENT TYPE	FD	RD	SS	PS	DS	UM	OM	MM	PT	RT
1	5%	5%	5%	5%	5%	5%	5%	10%	10%	10%
2	20	25	20	20	20	10	30	30	15	50
3	45	70	20	20	65	65	65	50	15	40
4	10	--	55	55	10	20	--	10	20	--
5	10	--	--	--	--	--	--	--	40	--
6	10	--	--	--	--	--	--	--	--	--

RELATIVE WEIGHT OF SECTIONS WITHIN DOCUMENT TYPES

Figure 4-1.

e. ST 18-151 sets forth the standards as set forth in DOD Standard 7935. Additional guidance is stated in TB 18-111. TB 18-111 should be referrd to before producing the document.

f. TB 18-111 refers to two additional documents that are not discussed in DOD Standard 7935. They are the Utility Software Manual (US) and the Utility Software Maintenance Manual (SM) (TEST). An alternative for the Program Maintenance Manual (MM) (TEST) is also presented.

SECTION 2. DOCUMENT TYPES

4.7. INTRODUCTION. This section contains an outline of the standards for each of the technical documents that may be produced during an ADS development effort. Each of the eleven document types is addressed separately, with the greatest amount of detail placed on the Functional Description. The entire table of contents for each document type, as it is prescribed by DOD Standard 7935, is given for each document, and a supplementary description is also included. (Refer to paragraph 4.7e and 4.7f on page 4.3, also).

4.8. Functional Description.

a. Preparation. An FD (Functional Description) is normally prepared for any system requiring a basis for mutual understanding between the Development Group and the User Group of a proposed ADS. It reflects the definition of the system requirements and provides the ultimate users with a clear statement of the operational capability to be developed. If the scope of the FD is changed at any point, the FD should be updated and receive user concurrence.

b. Uses. The FD is a tool for use by both computer and noncomputer oriented personnel and should be written as much as possible in noncomputer-oriented language, since many elements of the document will be subject to review by staff personnel who do not necessarily have a computer background.

FUNCTIONAL DESCRIPTION
TABLE OF CONTENTS

SECTION 1. GENERAL
- 1.1 Purpose of the Functional Description
- 1.2 Project References
- 1.3 Terms and Abbreviations

SECTION 2. SYSTEM SUMMARY
- 2.1 Background
- 2.2 Objectives
- 2.3 Existing Methods and Procedures
- 2.4 Proposed Methods and Procedures
- 2.4.1 Summary of Improvements
- 2.4.2 Summary of Impacts
  - 2.4.2.1 User Organization Impacts
  - 2.4.2.2 User Operational Impacts
  - 2.4.2.3 User Development Impacts
- 2.6 Assumptions and Constraints

SECTION 3. DETAILED CHARACTERISTICS
- 3.1 Specified Performance Requirements
- 3.1.1 Accuracy and Validity
- 3.1.2 Timing
- 3.2 Functional Area System Functions
- 3.3 Inputs-Outputs
- 3.4 Data Base Characteristics
- 3.5 Failure Contingencies
- 3.6 Security

SECTION 4. DESIGN DETAILS
- 4.1 System Description
- 4.2 System Functions
- 4.2.1 Accuracy and Validity
- 4.2.2 Timing
- 4.3 Flexibility
- 4.4 System Data
- 4.4.1 Inputs
- 4.4.2 Ouputs
- 4.4.3 Data Base

SECTION 5. ENVIRONMENT
- 5.1 Equipment Environment
- 5.2 Support Software Environment
- 5.3 Interfaces

Figure 4-2. Functional Description

5.4 Summary of Impacts
5.4.1 ADP Organization Impacts
5.4.2 ADP Operational Impacts
5.4.3 ADP Development Impacts
5.5 Failure Contingencies
5.6 Security
5.7 Assumptions and Constraints

SECTION 6. COST FACTORS

SECTION &. SYSTEM DEVELOPMENT PLAN

Figure 4-2. Functional Description

SECTION 1. GENERAL

1.1 Purpose of the Functional Description. This paragraph shall describe the purpose of the FD (Functional Description) in the following words, modified when appropriate:

This Functional Description for (Project Name) (Project Number) is written to provide:

a. The system requirements to be satisfied which will serve as a basis for mutual understanding between the user and the developer.

b. Information on performance requirements, preliminary design, and user impacts, including fixed and continuing costs.

c. A basis for the development of system tests.

1.2 Project References. This paragraph shall provide a brief summary of the references applicable to the history and development of the project. The general nature of the computer programs (tactical, war gaming, management information, etc.) to be developed shall be specified. The project sponsor, user, and operating center(s) that will run the completed computer programs shall be indicated.

At least the following documents[1] when applicable, shall be specified by author or source, reference number, title, date, and security classification.

a. Project request, a copy of which must be included as an appendix.

b. Previously developed technical documentation relating to this project.

c. Significant correspondence relating to the project to include formal agreements to the requirements contained in the Functional Description.

d. Documentation concerning related projects.

e. Other manuals or documents that constrain or explain technical factors affecting project development.

---

[1] When applicable, specific reference should be made to the provisions of these documents in subsequent sections of the Functional Description.

f. Standards or reference documents, such as:

(1) Documentation standards and specifications.

(2) Programming conventions.

(3) DOD or Federal standards (data element, programming languages, etc.).

(4) Hardware manuals, support system documentation, etc., if necessary, for an understanding of the FD.

1.3 Terms and Abbreviations. This paragraph shall provide a listing or include in an appendix any terms, definitions, or acronyms unique to this document and subject to interpretation by the user of the document. This listing will not include item names or data codes.

SECTION 2. SYSTEM SUMMARY

This section shall provide a general description, written in non-ADP terminology, of the existing system and of the requirements for the proposed ADS.

2.1 Background. Included within this paragraph, as necessary, will be any information concerning the background of the uses and purposes of the system to orient the reader. Reference must be made to higher order and parallel systems when needed to enhance the general description. The relationships between the project and other capabilities being developed concurrently shall be described.

2.2 Objectives. Statements of the major performance requirements or goals of the proposed computer program system must be included. These statements should be precise, quantified if possible, and may include examples. When applicable, related events, such as exercises or impending military operations, may be discussed. Any anticipated operational changes that will affect the system and its use shall be identified and the provisions within the system for including them shall be explained.

2.3 Existing Methods and Procedures. This paragraph shall provide a brief description of the current methods and procedures being employed to satisfy the existing information requirements. A chart must be provided depicting the existing data flow through the functional system from data acquisition through its processing and eventual output. This chart may be complemented by an explanation or another chart showing the sequence in which the operational functions are performed by the user and pointing out the support of those decision making activities that is provided by the present system. Additionally, at least the following information should be included in this description:

a. Organizational/personnel responsibilities.

b. Equipment being utilized.

c. Inputs and outputs including volume and frequency.

d. Deficiencies, including limitations, such as time delays.

2.4 Proposed Methods and Procedures. A description of the proposed methods and procedures shall be presented in this and the following paragraphs. This description, written in non-ADP terminology, should explain how the proposed system will interact with the functional processes of which the automated system will be supportive. When products from other functional or automated systems will be utilized with, or will become part of the proposed system, they will be referenced in this description.

A chart depicting the proposed data flow should be provided to present an overall view of the planned capabilities. If the proposed system eliminates or degrades any capabilities in the existing system, these capabilities must also be described as well as the reasons for their elimination or degradetion. Alternative methods and procedures that have been considered may be included. A chart showing the major functional processing steps and a chart showing the interacting organizations should be included within the following paragraphs wherever they best complement the narrative.

2.4.1 Summary of Improvements. This paragraph shall provide a qualitative and quantitative summary of the benefits to be obtained from the proposed system. A comparison to the deficiencies identified in paragraph 2.3 and the identification of any additional capabilities required, along with appropriate explanations, may be provided. Explicitly identified will be the required capabilities that will be satisfied by the proposed system. When improvements of the existing methods and procedures are a requirement, the extent of the anticipated improvements must be stated. The discussion shall include:

a. Functional improvements (new capabilities).

b. Improvements of degree (upgrading existing capabilities).

c. Timeliness (improved response time).

d. The elimination or reduction of existing capabilities that are no longer needed.

2.4.2 Summary of Impacts. This and the following paragraphs shall describe the anticipated impacts and associated costs of the proposed system on the existing organizational and operational environments of the user. Impacts on the user during the development of the system shall be noted.

2.4.2.1 User Organizational Impacts. Organizational impacts may include the modifications of responsibilities and the addition or elimination of responsibilities that will be necessary to use the proposed system. Any personnel eliminated will be identified and a discussion provided of the possibilities of retraining. Requirements for the number and skills of additional personnel will be identified. Included will be changes in authorized strength, location and position identifier, if known.

2.4.2.2 User Operational Impacts. The operational impacts on the organization during the use of the proposed system will be included. This discussion will consider the proposed interface between the user and the computer operating center; the impacts on the user to change from the current operational procedures; new data sources; quantity, type, and timeliness of data to be submitted for use in the system; data retention requirements; and modes of user operation based on peacetime, alert, and wartime conditions. Also included will be proposed methods for providing input data if these data are not already available.

2.4.2.3 User Development Impacts. Development impacts will include a discussion of all user efforts that will be required prior to implementation of the system, such as training, manpower required to develop or modify the data base, etc. This paragraph will also include any user requirement for the parallel operation of the new and existing system along with the potential impact on the user during the testing phase of the proposed system. Any additional activities to be provided by the user to aid development also will be included in this paragraph.

2.5 Assumptions and Constraints. This paragraph shall describe any user assumptions and constraints that will affect development and operation of the system. Any limitations affecting the desired capability (including the prediction of expected types of errors) and explicit identification of any desired capabilities that will not be provided by the proposed system shall be discussed. Examples of assumptions include organizational actions, budget decisions, operational environment or deployment requirements. Examples of constraints include operation environment, budget limitations, system implementation deadlines or regulatory policy.

SECTION 3. DETAILED CHARACTERISTICS

This section shall provide a detailed description of the performance requirements of the proposed ADS written in non-ADP terminology.

3.1 Specific Performance Requirements. This paragraph shall describe the specific performance requirements to be satisfied by the ADS. This presentation shall be a delineation of requirements on which the system design is to be based.2 The requirements will be stated in such a manner that system functions discussed in paragraph 3.2 and the system tests necessary for implementation can be related to them. A quantitative presentation of requirements will be included, such as the number of records that must be handled, maximum allowed time from query to receipt of the requested information, and flexibility required to accommodate changing user requirements.

3.1.1 Accuracy and Validity. This paragraph shall provide a description of the accuracy requirements placed upon the system. The following units must be considered:

a. Accuracy requirements of mathematical calculations.

b. Accuracy requirements of data.

c. Accuracy of transmitted data.

3.1.2. Timing. This paragraph shll provide a description of the timing requirements to be placed on the system. The following timing requirements shall be considered:

a. Response time from receipt of input data to availability of ADS products.

b. Response time to queries and to updates of data files.

c. Sequential relationship of functions.

d. Priorities imposed by types of inputs and changes in modes of operation.

e. Any deviations from specified response times for peak load periods, as applicable.

---

2 Anticipated deviations from any of the standards must be specified by the documents listed in the above paragraph 1.2 must be specifically indicated.

3.2 Functional Area System Functions. This paragraph shall amplify and describe by individual function the major functional processing steps contained in paragraph 2.4. This description should relate the functions to the performance requirements in paragraph 3.1.

3.3 Inputs-Outputs. This paragraph shall describe each data element in the data inputs to and outputs from the ADS. For each data element may be listed information such as the following:

a. Data element name.

b. Synonymous name.

c. Definition.

d. Format.

e. Range of values.

f. Unit of measurement.

g. Data item names, abbreviations and codes.

When the information is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dictionary. Any variations in either the inputs or outputs from the format or data item names tht will be used on the data base of the ADS as discussed in paragraph 3.4 must be specifically identified.

When available, the various input data formats shall be shown and the input medium (disk, cards, magnetic tape, analog-originated signals from revolving radar, etc.) shall be specified. When available, the various output data formats including any quality control outputs shall be shown and the output medium shall be specified. When possible these outputs should be related to the system functions described in paragraphs 3.2.

3.4 Data Base Characteristics. This paragraph shall provide a discussion concerning the data elements to be used in the data base. For each data element may be listed such information as:

a. Data element name.

b. Synonymous name.

c. Definition.

d. Format.

e. Range of values.

f. Unit of measurement.

g. Data item names, abbreviations, and codes.

When the information is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dictionary. An estimate of the data storage requirements in terms of size and number of records will be provided. A description of the expected growth of the data and related components should be provided.

3.5 Failure Contingencies. This paragraph shall provide a discussion of the alternative courses of action that may be taken to satisfy the information requirements if the proposed ADS fails. There shall be included as appropriate:

a. Back-up. A discussion shall be provided of the back-up requirements for ensuring the continued achievement of system functions given in paragraph 3.2. "Back-up" as used means the redundancy available in the event the primary system element goes down.

b. Fallback. An explanation of the fallback techniques for ensuring the continued satisfaction of the specific requirements of the system shall be provided. "Fallback" as used indicates the use of another system or other means to accomplish the system requirements. For example, the fallback technique for an automated system might be manual manipulation and recording of data.

3.6 Security. This paragraph shall describe the degree of sensitivity of the data, data files, inputs and outputs of the system. Consideration must be given to classified, personal, proprietary and management data.

SECTION 4. DESIGN DETAILS

This section shall provide a detailed description of how the proposed system will satisfy the functional requirements delineated in Sections 2 and 3.

4.1 System Description. This paragraph shall provide a general description of the proposed ADS. Related and interfacing systems and their documentation will be referenced as required to enhance this general description. Included within this description shall be a chart showing the relationship of the user organizations to the major components of the proposed ADS. This chart shall be based on the information included in paragraph 2.4.

4.2 System Functions. This paragraph shall describe the functions of the proposed ADS. There will be both quantitative and qualitative descriptions of how these functions will satisfy the requirements of paragraph 3.1. The functions must be described in such a manner that the system environment in Section 5 can be related to them.

4.2.1 Accuracy and Validity. This paragraph shall provide a description of the accuracy that will be achieved by the ADS. This description will be related to the requirements stated in paragraph 3.1.1. The following accuracy requirements must be considered:

a. Accuracy requirements of mathematical calculations.

b. Accuracy requirements of data.

c. Accuracy of transmitted data.

4.2.2 Timing. This paragraph shall provide a description of the timing considerations of the ADS. This description will be related to the requirements stated in paragraph 3.1.2. The following requirements must be considered:

a. Throughput time.

b. Response time to queries and to updates of data files.

c. Sequential relationship of system functions.

d. Priorities imposed by types of inputs and changes in modes of operation.

e. Timing requirements for the range of traffic load under varying operating conditions.

4.3 Flexibility. This paragraph shall provide a description of the capability to be incorporated for adapting the system to changing requirements such as anticipated operational changes, interaction with new or improved systems, and planned periodic changes. Components and procedures designed to be subject to change will be identified.

4.4 System Data. Included in this paragraph will be a description of the inputs, outputs, and data used. Each description will include the information below, if known.

4.4.1 Inputs.

a. Title and tag.

b. Format and acceptable range of values.

c. Number of items.

d. Means of entry and input initiation procedures, e.g., typewriter, card, tape, sensor, internal.

e. Expected volume and frequency, including special handling (such as queuing and priority handling) for peak load periods.

f. Priority, e.g., routine, emergency.

g. Sources, form at source, and disposition of source document.

h. Security considertions of input and individual items.

i. Requirements for timeliness.

4.4.2 Outputs.

a. Title and tag.

b. Format to include headings, line spacing, arrangement, totals, etc.

c. Number of items.

d. Preprinted form requirements.

e. Means of display, if known, e.g., CRT, printer, typewriter, projector, alarm type, internal.

f. Expected volume and frequency including special handling.

g. Priority, e.g., routine, emergency.

h. Timing requirements, e.g., response time.

i. Requirements for accuracy.

j. User recipients and use of displays, such as notification, trends or briefings.

k. Security considerations of output and individual items.

4.4.3 Data Base. Each data file, table, dictionary, or directory will be described as follows:

a. Title and tag.

b. Description of content.

c. Number of records or entries.

d. Storage media and size.

e. Security and considerations.

f. Data retention.

SECTION 5. ENVIRONMENT

This section shall describe the current ADP environment and project the environment needed to satisfy those requirements delineated in Sections 2 and 3.

5.1 Equipment Environment. This paragraph shall provide a description of the equipment capabilities required for the operation of the proposed ADS. This paragraph will present broad descriptions of the equipment presently available and the characteristics of any new equipment necessary based on the information in Section 3. A guideline for equipment to be described follows:

a. Processor(s), including number of each online/offline, and size of internal storage.

b. Storage media, including number of disk units, tape units, etc.

c. Output devices, including number of each online/offline.

d. Input devices, including number of each online/offline.

e. Communications net, including line speeds.

5.2 Support Software Environment. This paragraph shall provide a description of the support software with which the computer programs to be developed must interact. Included will be both support software, input and equipment simulators, and test software, if needed. The correct nomenclature, level (version), and documentation references of each such software system, subsystem, and program shall be provided. In addition, the language (compiler, assembler, program, query, etc.), the operating system, and any Data Management System to be used will be identified.

5.3 Interfaces. This paragraph shall provide a description of the interfaces with other systems and subsystems. For each interface, the following should be specified:

a. Description of operational considerations of data transfer, such as security considerations.

b. General description of data transfer requirements to and from the subject system and characteristics of communications media/systems used for transfer.

c. Format, unit of measurement, range of values, data codes.

d. Type of anticipated interface, such as manual or automatic.

e. Anticipated interface procedures, including telecommunication considerations.

5.4 Summary of Impacts. This paragraph shall describe the anticipated organizational, operational and development impacts of the proposed ADS on the ADP organizations.

5.4.1 ADP Organization Impacts. Organizational impacts may include the modifications of positional responsibilities and the addition or elimination of responsibilities that will be required by the proposed system. Any personnel interactions eliminated will be identified and a discussion provided of the possibilities or retraining. ADP personnel responsibilities will be discussed. Requirements for the number and skills of additional personnel will be identified. Included will be changes in authorized strength, location and position identifier, if known.

5.4.2 ADP Operational Impacts. This paragraph shall discuss impacts on the operational procedures of the data processing center(s) to implement the ADS. Included will be operational impacts caused by a change in equipment configuration, if known.

5.4.3 ADP Developement Impacts. This paragraph shall assess the personnel and ADP processing commitment necessary in the development and testing of the automated system. Additional requirements for program and data conversion will be addressed, if known, along with any additional development impacts resulting from the requirements in paragraph 2.4.2.3.

5.5 Failure Contingencies. This paragraph shall provide a discussion of possible failures of the hardware or software system, the consequences (in terms of system performance) of such failures, and the alternative courses of action that may be taken to satisfy the information requirements.

a. Restart. Include a discussion of the restart capabilities for ensuring effective and efficient recovery from a temporary problem within the hardware or software systems. The "restart" capability, as used, is a capability to resume operation from a point in the automated process prior to where the problem occurred, with appropriate restoration of data.

b. Other. The fallback and back-up contingencies described in paragraph 3.5 will be considered, as appropriate.

5.6 Security. This paragraph shall describe the degree of protection for the levels of availability, integrity and confidentiality that must be provided by the overall system and its components.

5.7 Assumptions and Constraints. This paragraph shall address any data automation assumptions and constraints that relate to development and operation of the automated system, as applicable.

SECTION 6. COST FACTORS

This section shall provide a summary of cost factors for the proposed system in accordance with DOD Instruction 7041.3, when applicable. While the proposed system responds directly to the project request, other factors may determine the need for this system, such as requirements of higher echelons of command, security considerations, telecommunications considerations, and the need to interface with other automated systems. General alternatives that may be discussed include those for system development and system design with consideration being given to equipment, software, supporting telecommunications requirements, organization, operation, etc. Reference may be made to such information contained in other documents.

SECTION 7. SYSTEM DEVELOPMENT PLAN

This section shall discuss the overall management approach to the development and implementation of the proposed computer system. Included may be a discussion of the documentation to be produced, time frames for the development of the system or the modules of the system, necessary liaison and participation by other organizations to ensure successful development, and any other factors that must be known prior to initiating development. Reference may be made to such information contained in other documents.

4.9 Data Requirements Data

a. Preparation. The RD (Data Requirements Document) is normally prepared when a data collection effort by the User Group is required to generate and maintain system files. The RD is a technical document. It is to be as detailed as possible concerning the definition of inputs required of the user; the procedures to be followed to provide this input to the system; the description of expected output data; the specification of all uses of standard data elements; and the data limitations of the system.

b. Data Elements. The term "data element" as used throughout this document includes its related features. The term refers to a data element or to its use in a data system, often called "data use identifier." The names and associated codes of many data elements have been standardized and shall be used to facilitate data exchange and commonality of data structures.

TABLE OF CONTENTS

SECTION 1. GENERAL
- 1.1 Purpose of the Data Requirements Document
- 1.2 Project References
- 1.3 Terms and Abbreviations
- 1.4 Modification of Data Requirements
- 1.5 Security

SECTION 2. DATA DESCRIPTION
- 2.1 Logical Organization of Static System Data
- 2.2 Logical Organization of Dynamic Input Data
- 2.3 Logical Organization of Dynamic Output Data
- 2.4 Internally Generated Data
- 2.5 System Data Constraints

SECTION 3. USER SUPPORT FOR DATA COLLECTION
- 3.1 Data Collection Requirements and Scope
- 3.2 Recommended Source of Input Data
- 3.3 Data Collection and Transfer Procedures
  - 3.3.1 Input Formats
  - 3.3.2 Output Formats
- 3.4 Data Base Impacts

Figure 4-3

SECTION 1. GENERAL

1.1 Purpose of the Data Requirements Document. This paragraph shall describe the purpose of the RD (Data Requirements Document) in the following words, modified when appropriate:

> The objectives of this Data Requirements Document for (Project Name) (Project Number) are to list and define data elements which the system must handle and to communicate data collection requirements to the user.

1.2 Project References. This paragraph shall provide a brief summary of the references applicable to the history and development of the project. The general nature of the computer programs (tactical, inventory control, war gaming, management information, etc.) to be developed shall be specified. The project sponsor, user, and operating center(s) that will run the completed ADS shall be indicated. A general statement shall be included of the scope of user support necessary to furnish the required system data. Included shall be a brief statement of the project objectives and a description of the relationship of this project to other projects. At least the following documents, when applicable, shall be specified, by author or source, reference number, title, date, and security classification:

a. Project request.

b. Other technical documentation.

c. Functional Decription.

1.3 Terms and Abbreviations. This paragraph shall provide a listing or include in an appendix any terms, definitions, or acronyms unique to this document and subject to interpretation by the user of the document. This listing will not include item names or data codes.

1.4 Modification of Data Requirements. This paragraph shall describe the procedures for implementing and documenting changes to the system data requirements, if applicable.

1.5 Security. This paragraph shall describe the degree of senstitivity of the data, data files, inputs and outputs of the system. Consideration must be given to classified, personal, proprietary and managment data. It shall also reflect the authority for the degree of sensitivity and for the levels of availability, integrity and confidentiality. Examples of authority sources for the degree of senstivity are classification authority for classified data, sensitivity measurement program and Federal Register reference for personal data, and sensitivity measurement algorithm for proprietary and other data. Examples of authority sources for the levels of availability, integrity and confidentiality are user, regulatory agency or federal statute.

SECTION 2. DATA DESCRIPTION

The data described in this section shall be separated into two categories, static data and dynamic data. Static data are defined as that data which are used mainly for reference during system operation and are usually generated or updated in widely separated time frames independent of normal system runs. Dynamic data include all data which are intended to be updated and which are input to a system during a normal run (including "real time" data such as targeting data) or are output by the system. Static data as described above are frequently referred to as parametric data and dynamic data as nonparametric data. Both, however, are composed of data elements. The data element names listed in paragraphs 2.1, 2.2 and 2.3 shall be the same as those contained in standard data element dictionaries, whenever applicable. Along with each data element name, either in the appropriate paragraph or summarized in some other place, shall be provided for each data element, as appropriate:

a. Synonymous name.

b. Definition.

c. Format.

d. Range of values.

e. Unit of measurement.

f. Data item names, abbreviations and codes.

When the information is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dictionary.

2.1 Logical Organization of Static System Data. This paragraph shall list the titles of static data elements used by the system for either parametric control or reference purposes. The data elements will be arranged in any logical groupings, such as functions, subjects, or other categories which are most relevant to their usage.

2.2 Logical Organization of Dynamic Input Data. This paragraph shall provide a list of the titles of those dynamic input data elements which constitute the data that are intended to be updated by a normal system run or during online operation. The data elements will be arranged in logical groupings, such as functions, subjects, or other categories which are most relevant to their usage.

2.3 Logical Organization of Dynamic Output Data. This paragraph shall provide a list of the titles of those dynamic output data elements which constitute the data that are intended to be updated by a normal system run or during online operation. The data elements will be arranged in logical groupings, such as functions, subjects or other categroies which are most relevant to their usage.

2.4 Internally Generated Data. This paragraph shall provide the internally generated data which are of interest to the user. The elements included in this paragraph need not be all inclusive, since they will be only of informational value to the user and will require no user action.

2.5 System Data Constraints. This paragraph shall contain only the known or anticipated system data constraints which have not been outlined in the FD. This information is to provide a general indication of the limits of the system with regard to future expansion or utilization; for example, a statement of the maximum size and the maximum number of files, records, and elements. Emphasis shall be placed on limits which could prove critical in future systems development.

SECTION 3: USER SUPPORT FOR DATA COLLECTION

All of the data elements required for the system will be listed in paragraphs 2.1, 2.2, and 2.3. This section will describe the data collection support necessary from the user to gather the data values for use by the particular system.

3.1 Data Collection Requirements and Scope. This paragraph shall describe the types of information required by the developer in order to establish the data values of each data element. It shall request, as a minimum, those types of information needed to describe the data element in accordance with the information required for the data element dictionary, such as the data element name, synonymous name, definition, format, range of values, unit of measurement, data item names, abbreviations and codes. When the informtion is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dictionary.

a. Input Source(s) of the Data Element. This item names the source(s) from which the data element will be fed into the system. The source or origin of the data will be, for example, an operator, station, organizational unit, etc., or its component group--not the hardware device used for entering the data. An exception can be made in the case of automatic sensor devices, but in this case, the device should be linked to another data element set, such as a vehicle carrying the device, organization responsible for its use or maintenance, geographic location, etc.

b. Input Device. This item names the hardware device used for entering the data into the system. In those cases where only certain special stations are the legal entry points, those discrete devices shall be specified.

c. Recipients. This item specifies those users (other than the originator) who should be cognizant of the data. In specifying recipients, distinctions should be made among the following classes of data elements:

(1) Data elements input to the system, processed by it, and output from it essentially unchanged.

(2) Data elements generated by a program and output to the user.

(3) Data elements that are input to the system but that are not output by it. In this case, the program would substitute for the user as the recipient.

d. Critical Value. Many elements that have a range of values will have one value that is particularly significant to the user. This may be a breakpoint, a minimum stock level, a critical wind velocity, etc. When applicable, the cirtical value and it significance to the user should be included.

e. Scales of Measurement. For numeric scales, if the successive steps are not equal to "one" on the units of measurement, then the increment shall be specified. For example, a data element representing pressure in pounds per square inch (the unit of measurement) may be incremented by pound, half-pound, or 10-pound increments. For numeric scales, the "scale zero" should be specified if it is not implicit in the unit of measurement; e.g., geometic mean, root mean square, harmonic mean) should be indicated. For nonnumeric scales any relationships indicated by the legal values should be stated if not otherwise specified. for example, a code indicating lubricant type by values A, B, C, etc. should show whether the value is arbitrary (A = paraffin oils, B = graphite, etc.) or whether it indicates the ordering of the lubricants by some parameter (A = viscosity less than SAE 10, B = viscosity SAE 10 through SAE 40, etc.).

f. Conversion Factors. Measured quantities that must go through analog and digital conversion processes shall have the conversion factors specified.

g. Output Form/Device. Output data elements may be presented to the user symbolically, graphically, audibly, or may be used as input to some other automated system. If the user will receive a sensible presentation, the description should specify whether the user will receive the data element as part of a hard copy printout, a symbol in a CRT display, a line on a drawing, a colored light, an alarm bell, etc. If the output is passed along to some other automated system, the medium shoiuld be described, e.g., magnetic tape, punched cards, an electronic signal to a solenoid switch.

h. Expansion Factors. For systems that are expected to under future expansion, the expansion factor to be added to the maximum number of entries of this data element should be included. For instance, if the maximum number of input devices is now 12, but is expected to be 96 three years from now, the 700% expansion factor should be specified.

i. Frequency of Update. Data elements that are input to the system or that are expected to be modified by the system on a periodic basis should have the frequency of update specified. If the input arrives in a random or in an "as occurred" manner, both the average frequency and some measure of the variance must be specified.

3.2 Recommended Source of Input Data. This paragraph shall delineate the source(s) of input data. Recommendations as to who shall be responsible for providing specific data inputs shall be stated here. This shall include recommendations regarding the establishment of a user input reporting organization, if required. Those data inputs dependent on interfacing systems, unrelated agencies, or specific documents should be the source that is delineated.

3.3 Data Collection and Transfer Procedures. Specific instructions for data collection procedures shall be given in paragraph 3.3. These instructions shall include detailed formats where applicable. Communications media and timing of inputs and outputs shall be stated.

3.3.1 Input Formats. This paragraph shall describe all input formats (card layouts, tape layouts, etc.) to include master file formats used by the system.

3.3.2 Output Formats. This paragraph shall describe all output formats (printer layout, CRT display frame, tape layout, etc.) created by the system.

3.4 Data Base Impacts. This paragraph shall describe the impacts associated with collection and maintenance of the data base on equipment, software, organizational, operational, and developmental environments. These descriptions will amplify paragraph 2.4.2 of the Functional Description. Impacts on the system resulting from deficiencies in the data base shall also be given.

4.10 System/Subsystem Spe

a. Preparation. An
may be prepared to guide t
breaks down readily into s
individual Subsystem Speci
logical breakdown of a sys
functions, where each brea
programs. If individual S
some point be bound togeth
System Specification may b
be broken down into smalle
into the lowest common den
be used to write a System

b. Uses. The System
prepared for systems perso
concerning the environment
guidance to the program de
subsystem interfaces. It
Specification will present
continuing design effort,
scope of the system effort
Sybsystem Specifications w
applicable to the particul

ification.

S (System Specification or Subsystem Specification)
e development of large projects. If the system
bsystems, this document may be used to prepare
ications. A subsystem is herein defined as the
em into separate areas of responsibility, such as
down is composed of a program or series of
bsystem Specifications are prepared, they may at
r to form a System Specification or a separate
written. Many systems, however, may not logically
components because they are already broken down
minator. In these cases this document outline may
pecification.

Subsystem Specification is a technical document
nel. It is to be as detailed as possible
and design elements in order to provide maximum
ign effort. This document also defines system/
s anticipated that the System/Subsystem
more detailed data than the FD as a result of the
ut it should be noted that any modification to the
should be submitted as changes to the FD.
ll consider only those segments of the FD that are
r subsystem.

SYSTEM/SUBSYSTEM SPECIFICATION
TABLE OF CONTENTS

SECTION 1. GENERAL
- 1.1 Purpose of the System/Subsystem Specification
- 1.2 Project References
- 1.3 Terms and Abbreviations

SECTION 2. SUMMARY OF REQUIREMENTS
- 2.1 System/Subsystem Description
- 2.2 System/Subsystem Functions
  - 2.2.1 Accuracy and Validity
  - 2.2.2 Timing
- 2.3 Flexibility

SECTION 3. ENVIRONMENT
- 3.1 Equipment Environment
- 3.2 Support Software Environment
- 3.3 Interfaces
- 3.4 Security
- 3.5 Controls

SECTION 4. DESIGN DETAILS
- 4.1 General Operating Procedures
- 4.2 System Logical Flow
- 4.3 System Data
  - 4.3.1 Inputs
  - 4.3.2 Outputs
  - 4.3.3 Data Base
- 4.4 Program Descriptions

SECTION 1. GENERAL

1.1 Purpose of the System/Subsystem Specification. This paragraph shall describe the purpose of the SS (System/Subsystem Specification) in the following words, modified when appropriate:

The System/Subsystem Specification for (Project Name) (Project Number) is written to fulfill the following objectives:

a. To provide a detailed definition of the system/subsystem functions.

b. To communicate details of the on-going analysis between the user's operational personnel and the appropriate developement personnel.

c. To define in detail the interfaces with other system and subsystem and the facilities to be utilized for accomplishing the interfaces.

1.2 Project References. This paragraph shall provide a brief summary of the references applicable to the history and development of the project. The general nature of the computer programs (tactical, inventory control, war gaming, management informaion, etc.) to be developed shall be specified. A brief description of the system shall include its purpose and uses. The project sponsor, user, and the operating center(s) that will run be completed computer programs shall be indicated. At least he following documents, when applicable, shall be specified by author or source, reference number, title, date, and security classification.

a. Functional Description.

b. Related System/Subsystem Specifications.

c. Any other pertinent documentation or significant correspondence not specified in the Functional Description.

1.3 Terms and Abbreviations. This paragraph shall provide a listing or include in an appendix any terms, definitions, or acronyms unique to this document or subject to interpretation by the user of the document. This listing will not include item names or data codes.

SECTION 2. SUMMARY OF REQUIREMENTS

This section shall provide a summary of the system characteristics and requirements. This section shall be an expansion of the information published in the FD to reflect the determination of additional details. Any changes to the characteristics and requirements set forth in Sections 2 and 3 of the FD must be specifically identified.

2.1 System/Subsystem Description. This paragraph shall provide a general description of the system/subsystem to establish a frame of reference for the remainder of the document. Higher order and parallel systems/subsystems and their documentation will be referenced as required to enhance this general description. Included within this description shall be a chart showing the relationship of the user organizations to the major components of the system and a chart showing the interrelationships of the system components for the system/subsystem. These charts shall be based on or be updated versions of the charts included in paragraph 4.1 of the FD. The more detailed charts to be included in Section 4 shall be based on the charts included in this paragraph.

2.2 System/Subsystem Functions. This paragraph shall describe the system/subsystem functions. There will be both qualitative an quantitative descriptions of how the system/subsystem functions will satisfy the requirements. Although the descriptions of the system/subsystem functions may be refined and more detailed as a result of the on-going analysis and design, they must maintain a direct relationship to the system functions established in paragraph 4.2 of the FD, and be stated in such a manner that the system/subsystem environment in Section 3 can be related to them.

2.2.1 Accuracy and Validity. This paragraph shall provide a description of accuracy requirements imposed on the system/subsystem. The requirements will be related to paragraph 4.2.1 of the FD. The following accuracy requirements must be considered:

a. Accuracy requirements of mathematical calculations.

b. Accuracy requirements of data.

c. Accuracy of transmitted data.

2.2.2 Timing. This paragraph shall provide a description of the timing requirements placed on the system/subsystem, if they are applicable. The requirements will be related to paragraph 4.2.2 of the FD. The following timing requirements may be considered:

a. Throughput time.

b. Response time to queries and to updates of data files.

c. Sequential relationship of system/subsystem function.

d. Priorities imposed by types of inputs and changes in modes of operation.

e. Timing requirements for the range of traffic load under varying operating conditions.

f. Sequencing and interleaving programs and systems (including the requirement for interrupting the operation of a program without loss of data).

2.3 Flexibility. This paragraph shall provide a description of the capability to be incorporated for adapting the system/subsystem to changing requirements, such as anticipated operational changes, interaction with new or improved systems, and planned periodic changes. Components and procedures designed to be subject to change will be identified.

SECTION 3. ENVIRONMENT

This section shall provide an expansion of the environment given in the FD to reflect the additional analysis and changes to the environment. Changes in the environment that do not affect he scope of the project as described in the FD and are the result of on-going analysis and design will be explicitly identified within the appropriate paragraphs of this section. These changes will be discussed in terms of the impacts on the currently available environmental components (equipment, software, etc.) as well as the impacts on estimates and functions which were based on the original planned environment.

3.1 Equipment Environment. This paragraph shall provide a description of the equipment required for the operation of the system/subsystem. Included will be descriptions of the equipment presently available as well as a more detailed discussion of the characteristics of any new equipment necessary. Equipment requirements will be related to the requirements stated in paragraph 5.1 of the FD. A guideline for equipment to be described follows:

a. Processor(s), including number of each onlone/offline and size of internal storage.

b. Storage media, including number of disk units, tape units, etc.

c. Input-output devices, including number of each online/offline.

d. Communications net, including line speeds.

3.2 Support Software Environment. This paragraph shall provide a description of the support software with which the computer programs to be developed must interact. Included will be both support software and test, if needed. The correct nomenclature and documentation references of each such software system, subsystem, and program shall be provided. Included must be a reference to the languages (compiler, assembler, program, query, etc.), the operating system, and any Data Management System (DMS) to be used. This description must relate to and expand on the information provided in paragraph 5.2 of the FD. If operation of the computer programs to be developed is dependent upon forthcoming changes to support software, the nature, status, and anticipated availability date of such changes must be identified and discussed.

3.3 Interfaces. This paragraph shall provide a description of the interfaces with other applications computer programs, including those of other operational capabilities and from other military organizations. The individual interfaces will be related to paragraph 5.3 of the FD. For each interface, the following shall be specified:

a. Type of interface, such as operator control of a terminal or program interfaces with other programs.

b. Description of operational implications of data transfer, including security considerations.

c. Data transfer requirements to and from the subject system/subsystem and characteristics of communications media/systems used for transfer.

d. Current formats of inerchanged data for the sending and receiving systems including data item names, codes, or abbreviations that are to be interchanged.

e. Interface procedures, including telecommunications considerations.

f. Interface equipment.

If Subsystem Specifications are being written, interfaces with other subsystems which are to be developed will be described in the same manner.

3.4 Security. This paragraph shall be related to paragraph 5.6 of the FD and shall reflect the levels of availability, integrity and confidentiality of the system and its components. The relationship of the components to each other, how each affects the other, and their sequence of execution shall be discussed, when applicable.

3.5 Controls. This paragraph shall provide a presentation of overall systems/ subsystem controls. Included in this paragraph will be controls such as record counts, accumulated counts, batch controls, etc. If no specific controls are to be established at the subsystem level, this will be stated.

SECTION 4: DESIGN DETAILS

4.1 General Operating Procedures. Included in ths paragraph will be a general description of the operating procedures such as load, start, stop, recovery, and restart.

4.2 System Logical Flow. This paragraph shall describe the logical flow of the system/subsystem. Logical flow of the system/subsystem will be presented primarily in the form of higher-level charts. A narrative presentation, when appropriate, will be used to supplement the chart. Charts will provide an integrated presentation of the system/subsystem dynamics, of entrances and exits, and of interfaces with other computer programs. Charts will effectively represent all modes of operations, priorities, cycles, and special handling. The charts will show general flow from input, through the system/subsystem, to the generation of output.

4.3 System Data. Included in this paragraph will be description of the inputs, outputs, and data base.[1] For each record type and its data elements, information such as the folowing will be provided. When the information is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dictionary.

4.3.1 Inputs. Each input will be described as follows:

a. Input Records.

(1) Title and tag.

(2) Sources, medium of source, and disposition of source medium.

(3) Expected volume and frequency, including special handling (such as queuing and priority handling) for high density periods.

(4) Priority, e.g., routine, emergency.

(5) The degree of sensitivity.

(6) Requirments for timeliness.

b. Input Data Elements.

(1) Name and tag.

---

[1] Optionally, inputs and ouputs may be described with the individual programs to which they relate.

(2) Position in the input record type.

(3) Synonymous names.

(4) Definition.

(5) Unit of measurement.

(6) Format and acceptable range of values.

(7) The degree of sensitivity.

(8) Data item names, abbreviations and codes including a specification as to which is used in the input.

4.3.2 Outputs. Each output will be described as follows:

a. Output Reports.

(1) Title and tag.

(2) Format, to include headings, line spacing, arrangement, totals, etc. A report layout sheet may be included.

(3) Means of display, if known, e.g., CRT, printer, typewriter, projector.

(4) Expected volume and frequency including special handling.

(5) Priority, e.g., routine, emergency.

(6) Timing requirements, e.g., response time.

(7) User recipients and use of displays, such as notification, trends, or briefings.

(8) Preprinted form requirements.

(9) The degree of sensitivity.

b. Output Data Elements.

(1) Name and tag.

(2) Synonymous name.

(3) Definition.

(4) Unit of measurement.

(5) Format and acceptable range of values.

(6) The degree of sensitivity.

(7) Data item names, abbrevations and codes including a specification as to which is used in the output.

4.3.3 Data Base. Each data base may be described as follows:

a. Summary.

(1) Title and tag.

(2) Narrative summary of content.

(3) Number of records or entries.

(4) Storage media and size.

(5) Degree of sensitivity.

(6) Retention schedule, including provisions for backup.

b. Data Base Data Elements.

(1) Name and tag.

(2) Position in the data base.

(3) Synonymous name.

(4) Definition.

(5) Unit of measurement.

(6) Format and acceptable range of values.

(7) The degree of sensitivity.

(8) Data item names, abbreviations and codes, including a specification as to which is used in the data base.

4.4 Program Descriptions. Paragraphs 4.4.1 through 4.4.n shall provide descriptions of the functions (related to paragraph 2.2) of the computer programs in the system/subsystem.

4.11. Program Specification.

a. Preparation. A PS (Program Specification) may be written after an SS to expand on its requirement or without any SS having been prepared. The PS may present modifications from the FD, but it should be noted that any modification to the scope of the system effort should be submitted as a change to the FD.

b. Uses. The PS is a technical document. The amount of detail to be included is dependent upon the use to be made of the docment within the program development. It is anticipated that the PS will present more detailed data than the FD and related SS as a result of the detailed program design effort. Furthermore, a PS will consider only those segments of an FD or SS that are applicable to the particular program.

PROGRAM SPECIFICATION
TABLE OF CONTENTS

SECTION 1. GENERAL
- 1.1 Purpose of the Program Specification
- 1.2 Project References
- 1.3 Terms and Abbreviations

SECTION 2. SUMMARY OF REQUIREMENTS
- 2.1 Program Descriptions
- 2.2 Program Functions
  - 2.2.1 Accuracy and Validity
  - 2.2.2 Timing
- 2.3 Flexibility

SECTION 3. ENVIRONMENT
- 3.1 Support Software Environment
- 3.2 Interfaces
- 3.3 Storage
- 3.4 Security
- 3.5 Controls

SECTION 4. DESIGN DETAILS
- 4.1 Program Operating Procedures
- 4.2 Inputs
- 4.3 Outputs
- 4.4 Data Environment
  - 4.4.1 Storage Allocation
  - 4.4.2 Data Retention
  - 4.4.3 Program Relationships
- 4.5 Program Logic

SECTION 1. GENERAL

1.1 Purpose of the Program Specification. This paragraph shall describe the purpose of the PS (Program Specification) in the following words or appropriate modifications thereto:

The objective of this Program Specification[1] for (Project Name) (Project Number) is to describe the program design in sufficient detail to permit program production by the programmer/coder.

1.2 Project References. This paragraph shall provide a brief summary of the references applicable to the history and development of the project. The general nature of the computer programs (tactical, inventory control, war gaming, management information, etc.) to be developed shall be specified. A brief description of the system shall include its purpose and uses. The project sponsor, user, and operating center(s) that will run the completed computer programs shall be included. At least the following documents, when applicable, shall be specified by author of source, reference number, title, date, and security classification:

a. Functional Description.

b. Associated System/Subystem Specifications.

c. Related Program Specifications.

d. Any other pertinent documentation or significant correspondence not specified in the Functional Description.

1.3 Terms and Abbreviations. This paragraph shall provide a listing or include in an appendix any terms, definitions or acronyms unique to this document and subject to interpretation by the user of the document. This listing will not include item names or data codes.

---

[1] A Program Specification may be prepared for each program in the system or one Program Specification may be prepared for a group of related programs within the system.

SECTION 2. SUMMARY OF REQUIREMENTS

2.1 Program Description. This paragraph shall provide a general description of the program to establish a frame of reference for the remainder of the document. It shall include a summary of the system requirements. The documentation of higher order systems/subsystems and related programs will be referenced, as required, to enhance the general description of the program. If the program is common to more than one system or subsystem, appropriate references will be made to the applicable SS and FD documents.

2.2 Program Functions. This paragraph shall describe the functions of the program. If a program does not in itself satisfy system or subsystem requirements, this fact shall be noted in the presentation with a statement showing how the aggregate of programs completely satisfies the functions. Although the descriptions of the functions may be refined and more detailed as a result of the on-going analysis and design, they must be directly related to the subsystem or system functions and they must be stated in such a manner that program environment can be related to the.

2.2.1 Accuracy and Validity. This paragraph shall provide a description of accuracy requirements imposed on the program. The requirements will be related to paragraph 4.2.1 of the FD or to paraagraph 2.2.1 of the SS. Changes in the accuracy requirements from the next higher order document will be explicitly identified. The following accuracy requirements must be considered:

a. Accuracy requirements of mathematical calculations.

b. Accuracy requirements of data.

c. Accuracy of transmitted data.

2.2.2 Timing. This paragraph shall provide a description of the timing requirements placed on the program, if they are applicable. The requirements will be related to paragraph 3.1.2 of the FD or to paragraph 2.2.2 of the SS. Changes in the timing requirements from the next higher order document will be explicitly identified. The followign timing requirements must be considered.

a. Throughput time.

b. Response time to queries and to updates of data files.

c. Sequential relationship of program functions and data flows.

d. Priorities imposed by types of inputs and changes in modes of operation.

e. Timing requirements for the range of traffic load under varying operating conditions.

f. Sequencing and interleaving programs and systems (including the requirements for interrupting the operation of a program without loss of data).

g. I/O transfer time required for disk, drum, tape, etc.

h. Internal processing time.

2.3 Flexibility. This paragraph shall provide a description of the capability to be incorporated for adapting the program to changing requirements, such as anticpated operational changes. interaction with new or improved programs, and planned periodic changes. Components and procedures designed to be subject to change wil be identified. This paragraph will be related to the portions of paragraph 2.3 of the SS that are applicable to the subject program.

SECTION 3. ENVIRONMENT

This section shall specify the current environment in which the system will operate indicating any changes made to the environment as reflected in the FD and SS.

3.1 Support Software Environment. This paragraph shall provide a description of the support software with which the computer programs to be developed must interact. Included will be both support software and test software, if needed. The correct nomenclature and documentation references of each such software system, subsystem, and program shall be provided. Included must be a reference to the languages (compiler, assembler, program, query, etc.) and the operating system to be used. This description must relate to and expand on the information provided in paragraph 5.2 of the FD and paragraph 3.2 of the SS. If the operation of the computer programs to be developed is dependent upon forthcoming changes to support software, the nature, status, and anticipated availability date of such changes must be identified and discussed.

3.2 Interfaces. This paragraph shall provide a description of the interfaces with other applications computer programs, including those of other operational capabilities and from other organizations. The individual interfaces will be related to paragraph 5.3 of the FD and paragraph 3.3 of the SS. For each each interface, the following shall be specified:

a. Type of interface, such as operator control of a terminal or program interfaces with other programs.

b. Description of operational implications of data transfer, including security considerations.

c. Data transfer requirements to and from the subject program (including data content, sequence, t)ming, format, volume, and processing), and characteristics of communications media/systems used for transfer.

d. Formats of data for both the sending and receiving systems including the data item names, codes or abbreviations that are to be interchanged.

e. Interface procedures, including telecommunications considerations.

f. Interface equipment.

g. Data conversion requirements.

3.3 Storage. This paragraph shall provide a description of storage requirements; use of internal storage and auxiliary storage such as tape, disk, drum and the estimated quantity of storage required for each. Each type of program must give consideration to the following types of information for the various storage media:

a. Internal storage, number of words/auxiliary core bank or bytes/auxiliary core bank, number of banks.

b. Drum storage, number uf words/field, number of fields/drum, number of drum assemblies.

c. Disk storage, number of words/zone or bytes/cyclinder, number of zones/disk or cylinders/disk, number of disks/disk unit, number of disk units.

d. Tape storage, number of adapters, number of tape drives/adapter, number of tapes.

In addition, the machine storage will be further allocated into permanent and temporary areas.

3.4 Security. This paragraph shall reflect the type and degree of sensitivity of the data, and the degree of sensitivity of the algorithjms. For all sensitive data, a statement shall be included as to whether the data are always sensitive become sensitive upon the occurrence of specific events, or change their degree of sensitivity upon the occurrence of specific events. This paragraph shall also specify the operational environment that must exist with the ADP activity when any sensitive component of the system is being processed.

3.5 Controls. This paragaph shall provide a presentation of any program controls. Included in this paragraph will be controls such as record counts, accumulated counts, batch controls, etc. These controls will be related to that portion of paragaph 3.5 of the SS that relates to the program being described. It no specific controls are to be established at the program level, this will be so stated.

SECTION 4. DESIGN DETAILS

4.1 Program Operating Procedures. Any special program requirements necessary for the implementation of system operating procedures shall be delineated. Also discussed shall be the interaction of all programs with executive support programs.

4.2 Inputs. This paragraph shall provide a detailed description of all computer program inputs. For each input type and its data elements information such as the following may be provided. When the information is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dicitionary. Examples of prepared inputs and input preparation forms may be included. Optionally, inputs may be described with the program or subprogram.

a. Input Records.

(1) Title and tag.

(2) Source, medium, and disposition of source medium.

(3) Expected volume and frequency including special handling.

(4) Priority, e.g, routine emergency.

(5) The degree of sensitivity.

(6) Disposition other than processing, such as logging, hard copy reproduction of input, storage location, and custodian.

(7) Flexibility, such as capability of omitting and adding items.

(8) Requirements for timeliness, throughput time, etc.

(9) Special handling, such as specification of special control cards.

b. Input Data Elements.

(1) Name and tag.

(2) Position in the input record type.

(3) Synonymous name.

(4) Definition.

(5) Unit of measurement.

(6) Formal and acceptable range of values.

(7) The degree of sensitivity.

(8) Data item names, abbreviations and codes, including a specification as to which is used in the input.

This paragraph shall include examples of prepared inputs and preparation forms, such as input creation sheets and communications message forms.

4.3 Outputs. This paragraph shall provide a detailed description of all computer program outputs. For each output type and its data elements information such as the following may be provided. When the information is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dictionary.

a. Output Reports.

(1) Title and tag.

(2) Format, to include headings, line spacing, arrangement, totals, etc. A report layout sheet may be included.

(3) Means of display, if known, e.g., CRT, printer, typewriter, projector.

(4) Expected volume and frequency, including special handling.

(5) Priority, e.g., routine, emergency.

(6) Timing requirements, e.g. response time.

(7) User recipients and use of displays, such as notification, trends, or briefings.

(8) Preprinted form requirements.

(9) The degree of sensitivity.

(10) Data selection criteria will be presented to establish the basis for selecting information for display, e.g., selection for a maintenance report may be on the basis of information in a file of aircraft type and aircraft usage.

(11) Disposition, including logging, film and hard copy printout reproduction and storage, numbers of copies required for distribution, place of storage, office responsible for permanent copy, retention period, and special handling required because of bulk, security considerations, and timing.

(12) Description of plots or graphic displays will include the coordinates to be used, symbols to be used, type of graphic technique (i.e., points or continuous), number of curves per sheet, etc.

(13) Explanation of symbols.

(14) Conditional and status indicators (code and definition).

(15) Supporting background information, e.g. maps.

b. Output Data Elements.

(1) Name and tag.

(2) Synonymous name.

(3) Definition.

(4) Unit of measurement.

(5) Format and acceptable range of values.

(6) The degree of sensitivity.

(7) Data item names, abbreviations and codes, including a specification as to which is used in the output.

4.4 Data Environment. This paragraph shall provide a description of the data base. When the information is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dictionary.

a. Summary.

(1) Title and tag.

(2) Narrative summary of content.

(3) Number of records of entries.

(4) Storage media and size.

(5) The degree of sensitivity.

(6) Relationship of each record to a common data base, if applicable.

b. Data Base Data Elements.

(1) Name and tag.

(2) Position in the data base.

(3) Synonymous name.

(4) Definition.

(5) Unit of measurement.

(6) Format and range of values.

(7) The degree of sensitivity.

(8) Data item names, abbreviations and codes including a specification as to which is used in the data base.

4.4.1 Storage Allocation. Storage allocation will be described for internal and auxiliary storage as follows:

a. Storage media.

b. Available storage on each medium.

c. Addresses of available storage.

d. Erasable working storage.

4.4.2 Data Retention. Data retention requirements will be described as follows:

a. Historic retention to include collection of data to be retained, format, storage medium, and time parameters.

b. Periodic report data, e.g., time retained after report generation and time retained to provide summary reports.

c. Summary report data, such as time retained after summary report.

4.4.3 Program Relationships. Interrelationships of a program with the data base(s) will be described to show those files and tables used in each program function.

4.5 Program Logic. This paragraph shall describe the logic of the program. Logical flow may be presented primarily in the form of charts. A narrative presentation, when appropriate will be used to supplement the charts. All charts will be keyed to the higher-order charts in the SS, if applicable, or the FD.

The logical flow shall provide a detailed description of the processing performed by the program. There shall be included for each program function noted in paragraph 2.2 a description of the operation of the program. All processes will be described to include algorithmic or logic data mainpulations and decision process involved. Conditions being tested for purposes of branching will be explained in detail, as well as methods for identifying error conditions and the resulting actions of the program. The charts and narrative will be related to the information in paragraph 4.2 of the SS, if applicable.

4.12 Data Base Specification.

a. Preparation. A DS (Data Base Specification) is generally prepared when many analysts/programmers will be involved in writing programs that will utilize the same data.

b. Uses. The DS is a technical document prepared for programmers. It shall be sufficiently detailed to permit program coding and data base generation by the Development Group. Since this document is intended to cover all types of systems, it does not make specific data or presentation formats mandatory. Developers of any given system are best qualified to devise the physical formats most useful and comprehensible to project personnel. However, to achieve consistency in documentation, the following practices shall apply in all DSs:

(1) Each graphic representation shall be followed by a narrative explanation.

(2) Each item of information shown in a graphic representation shall be consistent with standardized data element names as shown in data element libraries.

TABLE OF CONTENTS

SECTION 1. GENERAL
1.1 Purpose of the Data Base Specification
1.2 Project References
1.3 Terms and Abbreviations

SECTION 2. DATA BASE IDENTIFICATION AND DESCRIPTION
2.1 Data Base Identification
2.1.1 System Using The Data Base
2.1.2 Effective Dates
2.1.3 Storage Requirements
2.1.4 Physical Description of Data Base Files
2.2 Labeling/Tagging Conventions
2.3 Organization of the Data Base
2.4 Special Instructions
2.5 Support Programs Available for Handling the Data Base
2.6 Security

SECTION 3. DATA DEFINITIONS
3.1 Data Files
3.1.1 General Description of Data File
3.1.2 Physical Characteristics of Data Files
3.1.3 Logical Characteristics of Data Files
3.2 Tables
3.3 Items
3.4 Records and Entries

SECTION 4. INTEGRATED DATA BASE

SECTION 1. GENERAL

1.1 Purpose of the Data Base Specification. This paragraph shall describe the purpose of the DS (Data Base Specification) in the following words, modified when apppropriate:

The objectives of this Data Base Specification for (Project Name) (Project Number) are to describe the storage allocation and data base organization and to provide the basic design data necessary for the construction of the system files, tables, dictionaries, and directories.

1.2 Project References. This paragraph shall include a brief description of the system providing its purpose and use. It shall identify the project sponsor, user, and the operating center(s) that will run the completed computer programs. A list of applicable documents shall be included. At least the following documents, when applicable, shall be specified by author or source, reference number, title, date, and security classification:

a. Functional Description.

b. Data Requirements Document.

c. System/Subsystem Specification.

d. Program Specifications.

1.3 Terms and Abbreviations. This paragraph shall provide a listing or include in an appendix any terms, definitions, or acronyms unique to this dcoument and subject to interpretation by the user of the document. This listing will not include item names or data codes.

SECTION 2. DATA BASE IDENTIFICATION AND DESCRIPTION

This section shall provide all of the information necessary to identify and describe the data base being documented. It shall, in addition, contain various kinds of background information essential for proper utilization of the data base.

2.1 Data Base Identification. This paragraph shall give the code name, tag, or label by which each data base may be uniquely identified. Additional descriptive information shall also be given, whether or not it is implied in the identification code.

2.1.1 System Using the Data Base. The system of which this data base is a part shall be accurately and thoroughly identified. Included shall be the full system name; system code name; tag or label; and system model, modification, or version number. If more than one system uses this data base, each shall be identified.

2.1.2 Effective Dates. The first and last dates of the period during which this data base may be used with the above named system shall be given. The basis of the selection of the dates may vary. They may correspond to document publication dates, program testing periods, turnover periods, turnover or delivery dates, etc., depending upon the implementation plan being used for the system. This paragraph shall also indicate whether the data base is complete or incomplete, pre or post system delivery to the customer, experimental or permanent, and whether it supersedes or will be superseded by another data base. If it supersedes another data base, the availability date of the new data base must be specified.

2.1.3 Storage requirements. This paragraph shall contain the estimated internal and peripheral storage assignments for all of the programs and data of a system. In addition, it will contain information relevant to certain constraints and conditions under which the programs must operate. Such constraints, based upon a design for the most efficient utilization of the storage media, can apply to both autonomouly operated systems and systems that function within the environment of larger, machine operating systems.

2.1.4 Physical Description of Data Base Files. The physical characteristics of the master data base file(s) and duplicate working copies shall be given. This paragraph shall include the file media (disk, tape, or card deck), form of the file (symbolic or binary), and the respective codes used.

2.2 Labeling/Tagging Conventions. This paragraph shall discuss the system labeling/tagging conventions to the extent necessary for the programmer to use the conventions as a practical working tool. For example, specify the conventions used to identify new versions of the data base.

2.3 Organization of the Data Base. This paragraph shall provide system implementers with a single, central source of major design consideratons for the handling of the data base. The purpose of thi paragraph is to promote consistency of design concerning the organization and maniulation of the physical data base files. The following information shall be given for each kind of file media (tape, disk, etc.) containing the data base:

a. General file design and format.

b. Rationale of the design.

An example of the kind of information to be presented in this paragraph is shown in Figure 2-01 for a tape storage medium.

2.4 Special Instructions. This paragraph shall contain the instructions to be followed by all personnel who will contribute to the generation of the data base and who will use it for both testing and operational purposes. Such instructions may include:

a. Criteria for entering data into the data base.

b. Rules and procedures to be followed when submitting data for entry into the data base.

c. Identificaton of a data control unit, if applicable.

d. Formats for data description sheets and cards.

e. Machine run instructions for generating, modifying, updating or otherwise using the data base files. In very large systems, where the details of such instructions are extensive, this paragraph may reference sections of other programming documents (manuals or specifications) where this specific information can be found.

2.5 Support Programs Available for Handling the Data Base. In this paragraph, all of the support programs directly related to the data base shall be either referenced or, if required, discussed briefly. Descriptions shall include program name, functions, and major program operating considerations, such as operating time, hardware setup required, etc. The detailed program documentation shall also be cited. If the program is only referenced, the program name, document title, document number, and appropriate sections of the document shall be provided. Examples of such programs are:

a. Assemble Common Data Pool Programs (assemble symbolic data description informaton into binary system format and provide a common data pool for related system programs).

b. Data Base Analyzing Programs (reorganize, change, or compare data at the data-file and/or the record level).

2.3 Tapes

2.3.1 Environmental Data Tape.

a. File Design and Format. The Environmental Data Tape will contain three tape files as shown below:

```
                      TAPE ID LABEL

FILE 1         (                 )

                        EOF

               DIRECTORY OF REPORTS
               IN FILE 3
FILE 2
                        EOF

               REPORT FILE-2 RECORDS PER
               REPORT AS FOLLOWS:

FILE 3         REPORT 'A' CONTROL DATA

                     REPORT A

               REPORT 'B' CONTROL DATA

                     REPORT B

               ETC.
```

b. Design Rationale. All system tapes will contain an indentification label as the first tape file. All tape loading programs will check the labels against tape identification parameters input when the system is initialized.

The second tape file will contain the labels or tapes of all reports in the third file in their sequential order, and a core/peripheral storage indicator. The loader program will use this to locate reports on the tape and to construct a report storage directory table in core.

The third tape file will contain two records per report. The first will contain detailed control data about the report; the second will contain the binary report itself. The tape loader will move the reports to their assigned peripheral storage areas, and place the contents of the control records into the permanent directory maintained in core.

FIGURE 2-01. Example of Textual Description of a Data Base

c. Storage Allocation Programs (provide efficient facilities for modifying system control data and for maintaining sets of data at an organizational level).

d. Data Base Loading Programs (move or copy collections of data).

e. Information Processing nd Data Management systems (DMS, FIPS, NIPS).

Also to be included will be Electronic Accounting Machine (EAM) procedures created specifically to handled the data base.

2.6 Security. This paragraph shall contain an overview and discussion of the security considerations associated with the data of the overall system.

SECTION 3. DATA DEFINITIONS

This section shall include thorough, detailed definitions and descriptions of all of the data utilized by the system. The specific details of information required for each form of data may vary from system to system and will depend on the design characteristics of the operational or machine operating systems. The information specified below is intended to cover all kinds of systems and is not intended to be totally applicable to any system. In general, included is information about the physical data (as it is structured for direct use by the computer programs) and about the logical data represented by the physical data, i.e., the data as information regardless of how they are structured for use by the computer programs.

3.1 Data Files. For each data file information such as the following may be provided.

3.1.1 General Description of Data File.

a. Name and tag or label.

b. A brief statement of the purpose of the file and logical criteria used for its compilation.

c. The degree of sensitivity.

d. Conditions under which the file is modified or updated.

e. Restrictions and limitations on usage.

3.1.2 Physical Characteristics of Data Files.

a. File contents and format.

b. Primary and secondary storage media.

c. Form of the contents (symbolic, binary, mixed).

d. File control information used by programs, such as storage control items, directories, pointers, skip continue features, and end-of-file markers.

e. A representative of the file structure including tables, records, entries, items or other sub elements of the file.

3.1.3 Logical Characteristics of Data Files. Included in this paragraph shall be description of each data element in the data base. For each data element may be listed information such as the following:

a. Data element name.

b. Synonymous name.

c. Definition.

d. Format.

e. Range of values.

f. Unit of measurement.

g. Data item names, abbreviations and codes.

When the information is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dictionary. Any variations in either the inputs or outputs from the format or data items that will be used on the data base of the ADS must be specifically identified.

3.2 Tables. An adequate table definition shall contain the following information as applicable to the data base.

a. Table tag or label.

b. Full name or purpose of the table.

c. Data file containing the table.

d. Program subsystem that uses this table.

e. Logical divisions within table (internal table blocks or parts - not entries).

f. Basic table structure (fixed or variable length, fixed or variable entry structure).

(1) If fixed length table:

(a) Number of words/characters per entry.

(b) Number of entries in table.

(2) If variable length table:

(a) Maximum length (number of words/characters).

best copy available

(b) Maximum number of entries (if entry structure is fixed).

(c) Number of kinds of entries.

(d) Key and control items of each kind of entry.

(e) Indication of presence or absence of table control information.

(f) Other directly related tables plus a statement of the relationship, for example:

(g) Details of the structure of each entry type to include:

1 Contents and format of each entry type.

2 Tag or label of each item plus its entry position, for example, word/character number of bit position.

(h) Unique or significant characteristics, such as:

1 "To be loaded into hardware-protected registers only."

2 "Contents adapted to each installation" (for a multisite system).

3.3 Items. As used in this paragraph, the word "item" refers to a specific category of detailed information that has a defined positon within a table, or is a discrete data entity (simple item), for direct and immediate manipulation by a program. Used in this sense, the definition of an item is machine and program oriented rather than operationally oriented. Item definitions shall include, appropriate:

a. Tag or label.

b. Purpose of the item (a brief statement referenced to related data elements, when applicable).

c. The degree of sensitivity of the item and whether it maintains a specific degree of sensitivity or changes under certain conditions (e.g., removal or introduction of data files or specific data).

d. Table type in which it is found.

e. Position in table (word number and bit positions/level numbers/etc.).

f. Item use, e.g., table control item, entry structure key item, string control item, data item.

g. Item type, e.g., symbolic character, integer, fraction, mixed number (fixed or floating point), string, bead, status.

h. Item coding, depending upon the item type, for example:

(1) Symbolic - character code used.

(2) Integer - binary or binary coded decimal.

(3) Fraction - scaling factor.

(4) Mixed number, fixed point - point position.

(5) Status - the maximum number of conditions, form of status values (symbolic or numeric binary), a list of all acceptable status values or conditions.

i. Accessiblity factor - coded to indicate machine instruction modifiers that can expedite retrieving and storing of the item, e.g., FW (full word), LHW or RHW (left or right-hand word), B (byte size), M (mask necessary), etc.

3.4 Records and Entries. Frequently, the basic unit of the data file is a record or an entry. These data units shll be defined to the same degree as tables (see paragraph 3.2). In addition to the file structure presented in the language being used, the following information for each data unit shall be given:

a. Full name and purpose.

b. An explanation of each item.

c. Maximum size.

d. Graphic representation as it is contained in the file, and examples of the output obtained from the file with signifiant output editing features, if applicable.

SECTION 4. INTEGRATED DATA BASE

If the system will use an integrated data base, that integrated data base shall be identified in this section. Included within this section shall be a discussion of any impacts on the integrated data base of the data that are to be used in this system as described in Section 2. Such impacts may include recommended changes in the organization of the data in the integrated data base that would enhance system efficiency, the addition to the integrated data base of new data elements needed for this sytem, etc. Recommendations concerning changes in existing support software of the integrated data base, such as data mangement systems, should also be included. If the system data base is designed based on the assumption that planned or recommended enhancements to either the integrated data base or its support software will be available at the time of system implementation, these assumptions will be stated.

4.13 Users Manual. The primary purpose of the UM (Users Manual) is to serve the needs of the User Group. Sections 1 and 2 of the UM present general and specific information on a specific computer program system, and are directed toward an organization's general management and staff peronnel who have no need for detailed technical information concerning system implementaion or operation. Sections 3 and 4 of th UM address staff personnel, but are more detailed in the discussion about how to provide input to the system; respond to requests from the system for information; and make use of outputs from the system that may be in the form of hard copy, Cathode Ray Tube (CRT) displays, etc. Instructions for the operation of specific consoles or terminals may be included in Sections 5 and 6.

USERS MANUAL
TABLE OF CONTENTS

SECTION 1. GENERAL
- 1.1 Purpose of the Users Manual
- 1.2 Project References
- 1.3 Terms and Abbreviations
- 1.4 Security

SECTION 2. SYSTEM SUMMARY
- 2.1 System Application
- 2.2 System Operation
- 2.3 System Configuration
- 2.4 System Organization
- 2.5 Performance
- 2.6 Data Base
- 2.7 General Description of Inputs, Processing, Outputs

SECTION 3. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS
- 3.1 Initiation Procedures
- 3.2 Staff Input Requirements
  - 3.2.1 Input Formats
  - 3.2.2 Composition Rules
  - 3.2.3 Input Vocabulary
  - 3.2.4 Sample Inputs
- 3.3 Output Requirements
  - 3.3.1 Output Formats
  - 3.3.2 Sample Outputs
  - 3.3.3 Output Vocabulary
- 3.4 Utilization of System Outputs
- 3.5 Recovery and Error Correction Procedures

SECTION 4. FILE QUERY PROCEDURES
- 4.1 System Query Capabilities
- 4.2 Data Base Format
- 4.3 Query Preparation
- 4.4 Control Instructions

SECTION 5. TERMINAL DATA DISPLAY AND RETRIEVAL PROCEDURES
- 5.1 Available Capabilities
- 5.2 Data Base Format
- 5.3 Access Procedures
- 5.4 Display and Retrieval Procedures
- 5.5 Recovery and Error Correction Procedures
- 5.6 Termination Procedures

SECTION 6. TERMINAL DATA UPDATE PROCEDRES
- 6.1 Frequency
- 6.2 Restrictions
- 6.3 Sources
- 6.4 Access Procedures
- 6.5 Update Procedures
- 6.6 Recovery and Error Correction Procedures
- 6.7 Termination Procedures

SECTION 1. GENERAL

1.1 Purpose of the Users Manual. This paragraph shall describe the purpose of the UM (Users Manual) in the following words, modified when appropriate.

The objective of the Users Manual for (Project Name) (Project Number) is to provide the user's non-ADP personnel with the inFormation necessary to effectively use the system.

1.2 Project References. This paragraph shall provide a brief summary of the references aplicable to the history and development of the project. The general nature of the computer programs (tactical, inventory control, war-gaming, management information, etc.) developed shall be specified. A brief description of the system shall include its purposes and uses. Also indicated shall be the project sponsor and user as well as the operating center(s) that will run the completed computer programs. At least the following documents, when applicable, shall be specified by author or source, reference number, title, date, and security classification.

a. Project request.

b. Previously published documentation on the project.

c. Documentation concerning related projects.

d. Standards or reference documetation, such as:

    (1) Documentation standards and specifications.

    (2) Programming conventions.

    (3) DoD or Federal standards (data elements, programming languages, etc.).

1.3 Term and Abreviations. This paragraph shall provide a list or include in an appensix any terms, definitions or acronyms uiique to this document and subject to interpretation by the user of the document. This list will not include item names or data codes.

1.4 Security. This paragraph shall contain an overview and discussion of the security considerations associated with the data of the system.

SECTION 2. SYSTEM SUMMARY

2.1 System Application. The uses of the ADS in supporting the activities of the user's staff shall be generally stated and explained. The description shall include:

a. The purpose, reason, or rationale of the system.

b. Capabilities and operating improvements provided by the system.

c. Additional features, characteristics, and advantages considered appropriate in furnishing a clear, general description of the system and the benefits derived from it.

d. Functions performed by the system, such as preproccessing or postprocessing data input or output from a primary processor; maintenance of data files; display of submarine, surface or aircraft, etc.

2.2 System Operation. This paragraph will show the relationships of the functions performed by the system with the organizations or stations that are sources of input to the system and those that are recipients of output from it. Included shall be charts and a brief narrative description including only the who, what, where, and why concerning the inputs and outputs shown on the chart.

2.3 System Configuration. A brief narrative description of the equipment used by the system shall be given. It may include the type of computer and input and output devices.

2.4 System Organization. The objective of this paragraph shall be to present a general overview of the organization of the system. The presentation shall show, as appropriate, the logical parts of the system (such as subsystems and progams) and a brief description of their role in the operation of the system.

2.5 Performance. This paragraph shall present a brief description of the overall performance capabilities of the system, including how it meets the information requirements of the staff or how it supports associated activities. Performance measures and information of interest are represented by the following examples:

a. Input - types, volumes, rate of inputs accepted.

b. Output - types, volume, accuracy, rate of outputs that the system can produce.

c. Response time - include qualifications, where necessary, that affect response time in processing operational reports, such as listing a tape, compiling an object program, etc. Type and volume of input and equipment configuration are examples of items that may influence running time and, consequently, response time.

d. Limitations - for example, maximum size per unit of input, format constraints, restrictions on what data files may be queried and by what location, language constraints.

e. Error rate - capabilities for detecting various legal and logical errors and the means provided for error correction.

f. Processing time - show typical processing times.

g. Flexibility - note provisions allowing extension of the usage of the system.

h. Reliability - note system provisions that support, for example, alternate processing or a switch-over capability.

2.6 Data Base. The data files that are referenced, supported, or kept current by the system shall be identified in functional terms. The brief description should include the type of data in the file and the usage made of it. If the system does not have a file query capability as described in Section 4, this paragraph will include a description of the data elements included in the data base. For each data element may be listed information such as the following:

a. Data element name.

b. Synonymous name.

c. Definition.

d. Format.

e. Range of values.

f. Unit of measurement.

g. Data item names, abbreviations and codes.

When the information is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dictionary. Any variations in either the inputs or outputs from the format or data items that will be used on the data base of the ADS must be specifically identified.

2.7 General Description of Inputs, Processing, Outputs. This paragraph shll present a general narrative decription of the inputs, the flow of data through the processing cycle, and the resultant outputs.

a. Inputs. In describing the inputs, consideration should be given to the following:

(1) Purpose of input - explain why the input is made to the program system and note conditions or events requiring it submission.

(2) Content of input - describe what the input contains in the way of operational, control, or reference data.

(3) Associated inputs - describe any other inputs required by the system in addition to the direct input.

(4) Origin of inputs - identify the source or preparer of the input.

(5) Data files - identify in general or functional terms the data files associated with the input.

(6) Security considerations.

(7) Other - include additional remarks of general information.

b. Processing. In this paragraph, the relationship of the input to the output should be described with a general decription of the flow of data through the processing cycle.

c. Outputs. In describing the outputs, consideration should be given to the following:

(1) Output - list the outputs produced by the program system showing their relationship to the inputs.

(2) Purpose of the output - explain the reason for the output and note conditions or events that require its generation by the system.

(3) Content of output - describe in general terms the information provided by the output.

(4) Associated outputs - reference other system outputs that complement the information in this output.

(5) Distribution of outputs - note the recipients in the organization who receive this output.

(6) Security considerations.

(7) Other - describe additional items of general information.

SECTION 3. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

This section shall provide the details necessary to prepare staff inputs to the system. The logical arrangement of the information shall enable the staff and functional personnel to prepare required inputs. In addition, this section will explain in detail the characteristics and meaning of the information the processing system or an exclusively online system is being described, the following paragraphs should provide the necessary procedures for the staff to utilize the system. If an online system with batch processing capabilities is being described, this paragraph may reference the detail the procedures to be followed for the batch processing runs, or both may be presented herein. Optionally, the following information may be presented with each capability.

3.1 Initiation Procedures. The procedures that must be followed to initiate system operation will be detailed in this paragraph. Included may be information such as sample job request forms, sample control card formats or log-on procedures are standard or are detailed in another manual, that manual will be referenced.

3.2 Staff Input Requirements. The requirements to be observed in preparing entries to the program system shall be delineated in this paragraph for each different type or class of input. Typical considerations are the following:

a. Cause of input - note what operational conditions require the submission of the input (e.g., catastrophe, normal status report, need to enter parameters in a source program, need to update data, the desire to obtain particular data, the need to respond to a particular display).

b. Time of input - specify when the input must be prepared (e.g., periodically, randomly as a function of an operational situation).

c. Origin of input - identify the staff unit or station authorized to generate the input.

d. Medium of input - note the medium used to enter the input (e.g., keyboard, punched card, magnetic or paper tape).

e. Associated inputs - reference any related inputs that are required to be entered at the same time as this input.

f. Other - note any other applicable information, such as other recipients of the inputs; priority; security handling; variations on the basic input format using code or key indicators; limitations on what files may be interrogated by a particular type of input.

3.2.1 Input Formats. The layout form(s) used in the initial preparation of program system inputs shall be illustrated and the information which may be entered on the various sections and lines explainhed. The explanation of each entry provision shall be keyed to the sample form illustrated.

3.2.2 Composition Rules. This paragraph shall provide a description of the language and the grammatical rules and conventions that must be observed in order to prepare input that can be accepted by the program system. The rules of syntax, usage of punctuation, etc. will be explained. Items for consideration may include the followings:

a. Input length - e.g., 100 characters maximum.

b. Line length - e.g., 30 characters maximum.

c. Format - e.g., all input items must be left-justified.

d. Labeling - i.e., usage of tags or identifier to denote major data sets to the system.

e. Sequencing - i.e., the order and placement of items in the input.

f. Punctuation - i.e., spacing and use of symbols (virgule, asterisk, character combinations, etc.) to denote start and end of input, of lines, of data groups, etc.

g. Combination - i.e., rules forbidding use of particular character or parameter sets in an input.

3.3 Output Requirements. The requirements relevant to each class or type of output shall be described. Representative information that may be included for each class of output is:

a. Purpose - the reasons why the output is generated e.g., the desire to obtain particular data, due to the existence of an "exception" situation, to identify different operating units at different ranges.

b. Time - whether the output is randomly or periodically repoduced. If produced periodically, the period must be specified.

c. Options - any modifications or variations of the basic output that are available.

d. Media - physical form of the output, such as printout, CRT, tape, cards.

e. Location - where the output is required to appear, such as in the computer area or remotely at a paticular physical area or station.

f. Other - any additional requirements for this output, such as priority, security handling, asociated outputs that complement the information in this output output is presented shall be explained in detail. Explanations shall be keyed to particular parts of the format illustrated. Appropriate information that may be provided includes the following:

a. Header - the title, identification, time, number of output parts, and similar basic control data that may be contained in the header or control segment of the output shall be described.

b. Body - the information that may appear in the body or text of the output must be explained. Described shall be the significance of fixed data, such as columnar headings in tabular display types of output. The existence of subsets or sections in the output format (e.g., part A, part B) should be noted. In card/tape output, the position or column locations allocated to specific output information should be described.

c. Trailer - the control or reference information that may be appended to the body of information presented shall be discussed.

Additional characteristics concerning the make-up of outputs may include information such as the meanings of special symbols, etc.

3.3.2 Sample Outputs. Illustrations of the output obtainable from the system shall be given for each differenct class or type. The function or purpose of the output shall be explained. A detailed description including information such as the following may be provided:

a. Definition - the meaning and use of each information variable for the reader or user.

b. Source - item extracted from a specific input, from a data base file, calculated by system, etc.

c. Characteristics - concerning omissibility of the item under certain conditions of the output generation, range of values, unit of measure.

3.3.3 Output Vocabulary. Any codes or abbreviations that appear in the output in a form different from those used on the input described in paragraph 3.2.3 shall be describe in this paragraph.

3.4 Utilization of System Outputs. An explanation shall be given of the use of the output by the operational area or activity which receives it. For example, a summary report of POL (petroleum, oil, and lubricant) stocks may be received by a materiel control activity and, depending on the information in the report, action might be required to initiate the purchase or the report, action might be required to initiate the purchase or transfer of stocks to a particular location; the appearance of a blinking symbol on a CRT may require keyboard entries by several stations; etc.

3.5 Recovery and Error Correction Procedures. A list of the error codes generated by the application program and the corrective actions to be taken by the user to correct the condition shall be included within this paragraph. Also included in this paragraph shall be the procedures to be followed by the user to ensure that any recovery and restart capabiities can be utilized.

SECTION 4. FILE QUERY PROCEDURES

This section shall be prepared for those ADSs with a file query retrieval capability. The instructions necessary for recognition, preparation, and processing of a query applicable to the datra base shall be cited in detail. The descriptive techniques illustrated in paragraphs 4.1, 4.2, and 4.3 shall be utilized as applicable.

4.1 System Query Capabilitiy. This paragraph shall illustrate in tabular form the preprogrammed query capabilities provided by the system with a cross-reference to a query card format or query statement. An example is shown in Figure 4-01.

4.2 Data Base Format. This pararaph shall illustrate the data base format and content. An example is shown in Figure 4-02. If applicable, the format shall show both the data which are not subject to queries and the data which, even though not specifically requested, are extracted for some queries. For each data element may be listed information such as the following:

a. Data element name.

b. Synonymous name.

c. Definition.

d. Format.

e. Range of values.

f. Unit of measurement.

g. Data item names, abbreviations and codes.

When the information is published in a data element dictionary, reference to an entry in the dictionary will be made rather than including an extract from that dictionary. Any variations in either the inputs or outputs from the format or data items that are used on the data base must be specifically identified.

4.3 Query Preparation. Instructions shall be provided for the preparation of any necessary query title, request, and parameter input. Figure 4-03 shows an example of this format. The details of query input preparation in the context of each specific data base and system retrieval capability shall be repeated as necessary in the form of positive instructions. In cases when the retrieval capability is part of a support program system and query input formats are not needed, the specific query statement required shall be listed. Figure 4-04 shows a specific query statement. The formats provided will be used by control personnel to transcribe queries into the technical phrasing of the retrieval system.

4.4 Control Instructions. Instructions shall be provided for the control of the sequencing of runs and of the program necessary to extract the response to the query request from the data base. These instructions shall include the requirments for, and the preparation of, control cards which may be required by the system or application programs. If extensive information concerning control card prepartion is contained in support system documentation, this documentation may be referenced.

QUERY	QUERY CARD FORMAT
Numbers of employees within an organization	A
Number of employees in a specific pay grade	B
Total gross pay for employees within an organization	C
State tax year-to-date for a specific state	D
FICA tax year-to-date for a specific employee	E
Total deductions for a specific employee	F
Net pay for a specific employee	G

FIGURE 4-01. Relationship of Queries to Card Formats

FIGURE 4-01. Example of Preprogrammed Query Capability

ITEM NAME	RECORD POSITIONS	KIND OF DATA
ORG-NAME	1-30	Alpha-numeric
ORG-ID	31-36	Alpha-numeric
SOC-SEC-NO	37-45	Alpha-numeric
NAME	46-65	Alpha-numeric
PAY-GRADE	66-69	Alpha-numeric
GROSS-PAY	70-75	Signed-numeric
GROSS-PAY-YTD	76-83	Signed-numeric
FED-TAX	84-89	Signed-numeric
FED-TAX-YTD	90-97	Signed-numeric
FICA	98-103	Signed-numeric
FICA-YTD	104-111	Signed-numeric
STATE-TAX	112-117	Signed-numeric
STATE-TAX-YTD	118-125	Signed-numeric
STATE-TAX-CODE	126-127	Alpha-numeric
ALLOTMENTS	128-133	Signed-numeric
NET-PAY	134-139	Signed-numeric

Figure 4-02. Format of Data Record 1

Figure 4-02. Example of Data Record Format

FORMAT OF A QUERY CARD A (NUMBER OF EMPLOYEES WITHIN AN ORGANIZATION)

Query Item title	Char. Pos.	Content/Comments
Query Designator	1	Q Constant
File Number	2	01 Constant
Query Number	4	01 First Query
Security Classification	10	U Unclassified
Query Card Format Code	12	A
Organization	14	Insert ORG-ID as requested by query. Refer to data format for applicable code.

FIGURE 4-03. Format of Query Card A

FIGURE 4-03. Example of Query Card Format

INFORMATION PROCESSING SYSTEM

Request - No. of employees within an organization
(Office of Secretary of Defense)

Query Statement - IF ORG-ID EQ OSD LIST NO OF
EMPLOYEES

FIGURE 4-04. Query Statement

FIGURE 4-04. Example of Query Statement

SECTION 5. TERMINAL DATA DISPLAY AND RETRIEVAL PROCEDURES

5.1 Available Capabilities. The data display and retrieval capabilities available through terminal operations will be stated and explained in general terms.

5.2 Data Base Content. This paragraph will discuss the content and, if applicable, the format of the data base used by the system with emphasis on the relationships among the data that can be displayed or retrieved.

5.3 Access Procedures. Presented in this paragraph will be the sequence of steps required to access the data base. Included will be such information as the name of the system or subsystem being called and other control information such as access restrictions.

5.4 Display and Retrieval Procedures. Paragraphs 5.4.1 through 5.4.n will describe the step-by-step procedures necessary to produce the various displays and retrievals that are available through the use of a terminal.

5.5 Recovery and Error Correction Procedures. Error codes and messages should be provided indicating their meanings and any corrective actions that should be taken.

5.6 Termination Procedures. This paragraph will present the sequence of steps necessary to terminate the display or retrieval operation.

SECTION 6. TERMINAL DATA UPDATE PROCEDURES

6.1 Frequency. This paragraph will describe the frequency of data updates from terminals. Information such as the events that caused the update may be included.

6.2 Restrictions. This paragraph shall describe any restrictions on updating the data base. Included may be such factors as:

a. The offices or personel authorized to update.

b. Time periods wnen such updating is allowed.

c. Information for ensuring that only authorized updates are allowed.

6.3 Sources. Included in this paragraph will be a list of the sources used to obtain the data that will make up each update.

6.4 Access Procedures. Presented in this paragraph will be the sequence of steps required to access the data base. Included will be such information as the name of the system or subsystem being called and other control information such as access restrictions.

6.5 Update Procedures. Paragraph 6.5.1 through 5.6.n will provide information to enable an authorized user to update data in the system data base using a terminal. For reach type of update procedure information such as the name of the operation, input formats, and sample responses may be included.

6.6 Recovery and Error Correction Procedures. Error codes and messages should be provided indicating their meanings and any corrective actions that sould be taken. Any user initiated recovery procedures and validity checks should also be included in narrative form.

6.7 Termination Procedures. This paragraph shall present the step-by-step sequence of actions necessary to terminate the update.

4.14 Computer Operation Manua
precise and detailed informat
procedures necessary to succe
system. It is directed towar
responsible for the efficient
center. These readers are pr
external characteristics and
general, the manual shall be
an expository style in order
associated with the computer
concerned with the flow of in
present breakdowns or delinea
computer program.

. The OM (Computer Operation Manual) contains
i on the control requirements and operating
fully initiate, run, and terminate the subject
supervisory and operator personnel who are
erformance of their organization's computer
arily interested in detailed information on the
erating procedures of a computer program. In
itten in a step-by-step fashion as opposed to
clarify and emphasize the procedures
rograms. Supporting illustratioi shall be
ut data and output informatioibut shall not
ions of the internal logic and flows within a

TABLE OF CONTENTS

SECTION 1. GENERAL
1.1 Purpose of the Computer Operation Manual
1.2 Project References
1.3 Terms and Abbreviations

SECTION 2. SYSTEM OVERVIEW
2.1 System Application
2.2 System Organization
2.3 Program Inventory
2.4 File Inventory
2.5 Processing Overview
2.6 Security

SECTION 3. DESCRIPTION OF RUNS
3.1 Run Inventory
3.2 Phasing
3.3 Run Description (Identify)
3.3.1 Control Inputs
3.3.2 Management Information
3.3.3 Input-Output Files
3.3.4 Output Reports
3.3.5 Reproduced Output Reports
3.3.6 Restart/Recovery Procedures
3.4 Run Description (Identify)

SECTION 1. GENERAL

1.1 Purpose of the Computer Operation Manual. This paragraph shall describe the purpose of the OM (Compter Operation Manual) in the following words or appropriate modifications thereto:

> The objective of this Computer Operation Manual for (Project Name) (Project Number) is to provide computer control and computer operator personnel with a detailed operational description of the system and its associated environment with which they will be concerned during the performance of their duties.

1.2 Project References. At least the following docments, when applicable, shall be specified by author or source, reference number, title, date, and security classification:

a. Users Manual.

b. Program Maintenance Manual.

c. Other pertinent documentaton on the project.

1.3 Terms and Abbreviations. This paragraph shall provide a listing or include in an appendix any terms, definitions, or acronyms unique to this document and submect to interpretation by the user of the document. This listing will not include item names or data codes.

SECTION 2. SYSTEM OVERVIEW

2.1 System Application. A brief description of the system including its purpose and uses shall be provided.

2.2 System Organization. This paragraph shall describe the operation of the system by use of a chart showing the data processing operations, including how the different operations are interrelated. If sets of runs are grouped by time periods or cycles, then each set of integrated operations required on a daily, weekly, etc. basis will be presented. If runs may be grouped logically by organizational level, the groups of runs that can be performed by each organizational level such as headquarters processing, field activity processing, etc., shall be presented.

2.3 Program Inventory. This paragraph shall provide an inventory of the various programs. This listing shall include the program full name, program ID, as well as security considerations of the programs and identification of those programs necessary to continue or resume operation of the ADS in case of an emergency.

2.4 File Inventory. This paragraph shall list all permanent files that are referenced, created, or updated by the system. This listing shall include information such as the file name, file ID, storage medium and required storage (number of tapes or disks) as well as security considerations. The listing shall also identify those files necessary to continue or resume operation of the ADS in case of an emergency.

2.5 Processing Overview. This paragraph will provide information which is applicable to the processing of the system. Separate paragraphs may be used as needed to cover system restrictions, waivers of operational standards, information oriented toward specific support areas (e.g., library, EAM support) or other processing requirements such as the following:

a. Interface with other systems.

b. Other pertinent system-related information.

2.6 Security. This paragraph shall contain an overview and discussion of the security considerations associated with the data of the system.

SECTON 3. DESCRIPTION OF RUNS

This section shall provide a description of the runs for operations and scheduling personnel to allow accurate and efficient scheduling of operations, assignment of equipment, the management of input and output data, and restart/recovery procedures. In online systems some information about system operational control will be related to the capabilities of the operating system and other information will need to be presented in a manner more directly useful to operators of online terminals. Much of the necessary information should be included in figures with additional information that is specifically oriented to the hardware and software set being used.

3.1 Run Inventory. This paragaph shall provide a list of the various runs (i.e., programs, jobs) that may be made by the system and include a brief summary of the purpose of the run. This list should relate to the runs that are included in the remainder of this section and should show the programs that are executed during the run.

3.2 Phasing. This paragraph shall provide a schedule of acceptable phasing of the program system into a logical series of operations. A system run may be phased to permit manual or semiautomatic checking of intermediate results, to permit a logical break if higher priority jobs are submitted. An example of the minimum division for most systems would be edit, file update, and report preparation.

3.3 Run Description (Identify). Paragraph 3.3 through 3.n will provide the detailed information needed to execute runs of the system. The information provided will be organized in a manner most useful to the operating centers and operations personnel that will perform the runs.

3.3.1 Control Inputs. This paragraph shall provide a listing of the runstream of job control statements needed to initiate the run.

3.3.2 Management Information. This paragraph shall present the information needed to manage the run including, for example, the following information:

a. Run identification.

b. Peripheral and resource requirements.

c. Security considerations.

d. Method of initiation, such as on request, as a result of another run, at a predetermined time, etc.

e. Estimated run time.

f. Required turnaround time.

g. Messages and responses.

h. Procedures for taking check points.

i. Waivers from operational standards.

j. Contacts for problems experienced with the run.

3.3.3 Input-Output Files. This paragraph shall list information about the files that serve as input to or that are created or updated by the run. Included for each file should be information such as the following:

a. File name.

b. Security and privacy.

c. Recording medium.

d. Retention schedule.

e. Disposition of file.

3.3.4 Output Reports. This paragraph shall list information about the reports that are produced druing the run. Included for each report should be information such as the following:

a Report identification.

b. Security and privacy.

c. Medium (i.e., hardcopy, tape).

d. Volume of report.

e. Number of report.

e. Number of copies.

f. Distribution of copies.

3.3.5 Reproduced Output Reports. this paragraph shall provide information about those competer-generated reports that are subsequently reproduced by other means. Included for each report shall be information such as the following:

a. Report identification.

b. Security and privacy.

c. Reproduction technique.

d. Paper size.

e. Binding method.

f. Number of copies.

g. Distribution of copies.

3.3.6 Restart/Recovery Procedures. This paragraph shall provide information to the operations center personnel concerning restrart/recovery procedures that these personnel will follow in the event of a system failure.

3.4 Run Description (Identify). Paragraph 3.4 will present information about the second run in the manner similar to that used in paragraph 3.3.

4.15 Program Maintenance Manual. The MM (Program Maintenance Manual) presents general and specific information on the computer program. It is written for personnel who are responsible for the maintenance of the computer programs. It will describe the computer programs in a detailed, technical presentation to assist the maintenance programmer.

PROGRAM MAINTENANCE MANUAL
TABLE OF CONTENTS

SECTION 1. GENERAL DESCRIPTION
- 1.1 Purpose of the Program Maintenance Manual
- 1.2 Project References
- 1.3 Terms and Abbreviations

SECTION 2. SYSTEM DESCRIPTION
- 2.1 System Application
- 2.2 Security
- 2.3 General Description
- 2.4 Program Description

SECTION 3. ENVIRONMENT
- 3.1 Equipment Environment
- 3.2 Support Software
- 3.3 Data Base
  - 3.3.1 General Characteristics
  - 3.3.2 Organization and Detailed Description

SECTION 4. PROGRAM MAINTENANCE PROCEDURES.
- 4.1 Conventions
- 4.2 Verification Procedures
- 4.3 Error Conditions
- 4.4 Special Maintenance Procedures
- 4.5 Special Maintenance Programs
- 4.6 Listings

SECTION 1. GENERAL

1.1 Purpose of the Program Mainetenance Manual. This paragraph shall describe the purpose of the MM (Program Maintenance Manual) in the following words or appropriate modifications thereto:

The objective for writing this Program Maintenace manual for (Project Name) (Project Number) is to provide the maintenace programmer personnel with the information necessary to effectively maintain the system.

1.2 Project References. This paragraph shall provide a brief summary of the references applicable to the history and development of the project. The general nature of the system (tactical, inventory control, war-gaming, management information, etc.) developed shall be specified. A brief description of this system shall include its purpose and uses. Also indicated shall be the project sponsor and user as well as the operating center(s) that wil) run the completed computer programs. At least the following documents, when applicable, shall be specified by author or source, reference number, title, date, and security classification:

a. Users Manual.

b. Computer Operation Manual.

c Other pertinent documentation on the project.

1.3 Terms and Abbreviations. This paragraph shall provide a list or include in an appendix any terms, definitions or acronyms unique to this document and subject to interpretation by the user of the document. This list will not include item names or data codes.

SECTION 2. SYSTEM DESCRIPTION

2.1 System Application. The purpose of the system and the functions it performs shall be explained. A particular application system, for example, might serve to control mission activities by accepting specific inputs (status reports, emergency conditions), extracting items of data, and deriving other items of data in order to produce both information about a specific mission and information for summary reports. These functions shall be related to paragraphs 3.1, Specific Perfromance Requirements, and 4.2, System Functions, of the FD.

2.2 Security. This paragraph shall contain an overview and discussion of the security considerations associated with the data of the system.

2.3 General Description. This paragraph will provide a comprehensive description of the system, subsystem, jobs, etc. in terms of their overall functions. This description will be accompanied by a chart showing the interrelationships of the major components of the system.

2.4 Program Description. The purpose of this paragraph is to supply details and characteristics of each program and subroutine that would be of value to a maintenance programmer in understanding the programs related to the specific system being documented will be discussed under paragraph 4.4, Special Maintenance Procedures.) This paragraph will initially contain a list of all programs to be discussed, followed by a narrative description of each program and its respective subroutines under separate paragraphs starting with 2.4.1 through 2.4.n. For each major item listed below include any applicable information on security considerations. Information to be included in the narrative description is represented by the following items:

a. Identification - program title or tag, including a designation of the version number of the program.

b. Functions - description of program functions and the method used in the program to accomplish the function.

c. Input - description of the input. Description used here must include all information pertinent to maintenance programming including:

(1) Data records used by the program during operation.

(2) Input data type and location(s) used by the program when its operation begins.

(3) Entry requirements concerning the initiation of the program.

d. Processing - description of the processing performed by the program, including:

(1) Major operations - major operations of the program will be described. The description may reference chart(s) which may be included in an appendix. This chart will show the general logical flow of operations, such as read an input, access a data record, major decision, and print an output which would be represented by segments or subprograms within the program. Reference may be made to include charts that present each major operation in more detail.

(2) Major branching conditions provided in the program.

(3) Restrictions that have been designed into the system with respect to the operation of this program, or any limitations on the use of the program.

(4) Exit requirements concerning termination of the operation of the program.

(5) Communications or linkage to the next logical program (operational, control).

(6) Output data type and location(s) produced by the program for use by related processing segments of the system.

(7) Storage - Specify the amount and type of storage required to use the program and the broad parameters of the storage locations needed.

e. Output - description of the outputs produced by the program. While this description may reference output described in the Users Manual, any intermediate output, working files, etc., should be described for the benefit of the maintenance programmer.

f. Interfaces - description of the interfaces to and from this program.

g. Tables and Items - provide details and characteristics of the tables and items within each program. Items not part of a table must be listed separately. Items contained within a table may be referenced from the table descripitons. If the data description of the program provides sufficient information, the program listing may be referenced to provide some of the necessary information. At least the following will be included for each table:

(1) Table tag, label or symbolic name.

(2) Full name and purpose of the table.

(3) Other programs that use this table.

(4) Logical divisions within the table (internal table blocks or parts - not entries).

(5) Basic table structure (fixed or variable length, fixed or variable entry structure).

(6) Table layout (a graphic presentation should be used). Included in supporting description should be table control information, details of the structure of each type of entry, unique or significant characteristics of the use of the table, and information about the names and locations of items within the table.

(7) Item - the term "item" refers to a specific category of detailed information that is coded for direct and immediate manipulation by a program. Used in this sense, the definition of an item is machine and program oriented rather than operationally oriented. Of primary importance is an explantaion of the use of each item. At least the following will be included for each item:

(a) Item tag or label and full name.

(b) Purpose of the item.

(c) Item coding, depending upon the item type, such as integer, symbolic, status, etc.

h. Run Features - description of any unique features of the running of this program that are not included in the OM.

SECTION 3. ENVIRONMENT

3.1 Equipment Environment. This paragraph shall discuss the equipment configuration and its general characteristics as they apply to the system.

3.2 Support Software. This paragraph shall list the various support software used by the system and identify the version or release number under which the system was developed.

3.3 Data Base. Information in this paragraph shall include a complete description of the nature and content of each data base used by the system including security considerations.

3.3. General Characteristics. Provide a general description of the characteristics of the data base, including:

a. Identification - name and mnemonic reference. List the programs utilizing the data base.

b. Data Permanency - note whether the data base contains static data that a program can reference, but may not change, or dynamic data that can be changed or updated during system operation. Indicate whether the change is periodic or random as a function of input data.

c. Storage. specify the storage media for the data base (e.g., tape, disk, internal storage) and the amont of storage required.

d. Restrictions - explain any limitations on the use of this data base by the program in the system.

3.3.2 Organization and Detailed Description. This paragraph will serve to define the internal structure of the data base. A layout will be shown and its composition, such as records and tables, will be explained. If available, computer-generated or other listings of this detailed information may be referenced or included, herein. The following items indicate the type of information desired:

a. Layout - shown the structure of the data base including records and items.

b. Sections - note whether the physical record is a logical record or one of several that constitute a logical record. Identify the record parts, such as header or control segment and the body of the records.

c. Fields - identify each field in the record structure and, if necessary, explain its purpose. Include for each field the following items:

(1) Tags/labels - indicate the tag or label assigned to reference each field.

(2) Size - indicate the length and number of bits/characters that make up each data field.

(3) Range - indicate the range of acceptable value for the field entry.

d. Expansion - note provisions, if any, for adding additional data fiel to the record.

SECTION 4. PROGRAM MAINTENANCE PROCEDURES

This section shall provide information on the specific procedures necessary for the programmer to maintain the programs that make up the system.

4.1 Conventions. This paragraph will explain all rules, schemes, and conventions that have been used within the system. Information of this nature could include the following items.

a. Design of mnemonic identifiers and their application to the tagging or labeling of programs, subroutines, records, data fields, storage areas, etc.

b. Procedures and standards for charts, listings, serialization of cards, abbreviations used in statements and remarks, and symbols appearing in charts and listings.

c. The appropriate standards, fully identified, may be reference in lieu of a detailed outline of conventions.

d. Standard data elements and related features.

4.2 Verification Procedures. This paragraph will include those requirements and procedures necessary to check the performance of a program section following its modification. Included may also be procedures for periodic verification of the program.

4.3 Error Conditions. A description of error conditions, not previously documented, may also be included. This description shall include an explanation of the source of the error and recommended methods to correct it.

4.4 Special Maintenance Procedures. This paragraph shall contain any special procedures required which have not been delineated elsewhere in this section. Specific information that may be appropriate for presentation would include:

a. Requirements, procedures, and verification which may be necessary to maintain the system input-output components, such as the data base.

b. Requirements, procedures, and verification methods necessary to perform a Library Maintenance System run.

4.5 Special Maintenance Programs. This paragraph shall contain an inventory and description of any special programs (such as file restoration, purging history files) used to maintain the system. These programs should be described in the same manner as those described in paragraphs 2.3 and 2.4 of the MM.

a. Input-Output Requirements. Included in this paragraph shall be the requirements concerning materials needed to support the necessary maintenance tasks. Materials may, for example, include card decks for loading a maintenance program and the input which represent the changes to be made. When a support system is being used, this paragraph should reference the appropriate manual.

b. Procedures. The procedures, presented in a step-by-step manner, shall detail the method of preparing the inputs, such as structuring and sequencing of inputs. The operations or steps to be followed in setting up, running, and terminating the maintenance task on the equipment shall be given.

4.6 Listings. This paragraph will contain or provide a reference to the location of the program listing. Comments appropriate to particular instruction shall be made if necessary to understand and follow the listing.

4.16 Test Plan. The PT (Test P
and contains the orderly schedul
effect a comprehensive test of a
directed toward the user staff p
language and those the operation
terminology.

an) is a tool for directing the ADS testing, of events and list of materials necessary to complete ADS. Those parts of the document rsonnel shall be presented in nontechnical personnel shall be presented in suitable

TEST PLAN
TABLE OF CONTENT

SECTION 1. GENERAL INFORMATION
- 1.1 Purpose of the Test Plan
- 1.2 Project References
- 1.3 Terms and Abbreviations

SECTION 2. DEVELOPMENT TEST ACTIVITY
- 2.1 Statement of PreTest Activity
- 2.2 Pre-Test Activity Results

SECTION 3. TEST PLAN
- 3.1 System Description
- 3.2 Testing Schedule
- 3.3 First Location (Identify) Testing
  - 3.3.1 Milestone Chart
  - 3.3.2 Equipment Requirements
  - 3.3.3 Software
  - 3.3.4 Personnel
  - 3.3.5 Orientation Plan
  - 3.3.6 Test Materials
    - 3.3.6.1 Deliverable Material
    - 3.3.6.2 Site Supplied Materials
  - 3.3.7 Security
- 3.4 Second Location (Identify) Testing

SECTION 4. TEST SPECIFICATION AND EVALUATION
- 4.1 Test Specification
  - 4.1.1 Requirements
  - 4.1.2 System Functions
  - 4.1.3 Test/Function Requirements
- 4.2 Test Methods and Constraints
  - 4.2.1 System Test Conditions
  - 4.2.2 Extent of System Test
  - 4.2.3 Data Recording
  - 4.2.4 System Test Constraints
- 4.3 Test Progression
- 4.4 Test Evaluation
  - 4.4.1 Test Data Criteria
  - 4.4.2 Test Data Reduction

SECTION 5. TEST (IDENTIFY) DESCRIPTIONS
5.1 Test Description
5.2 Test Control
5.2.1 System Test Means of Control
5.2.2 Test Data
5.2.2.1 Input Data
5.2.2.2 Input Commands
5.2.2.3 Output Data
5.2.2.4 Output Notification
5.3 Test Procedures
5.3.1 Test Setup
5.3.2 System Initialization
5.3.3 Test Steps

SECTION 1. GENERAL

1.1 Purpose of the Test Plan. This paragraph shall describe the purpose of the PT (Test Plan) in the following words, modified as appropriate:

The Test Plan for (Project Name) (Project Number) is written to fulfill the following objectives:

a. To provide guidance for the management and technical effort necessary throughout the test period.

b. To establish a comprehensive test plan and to communicate to the user(s) the nature and extent of the tests deemed necessary to provide a basis for evaluation of the system.

c. To coordinate with the user an orderly schedule of events, a specification of equipment and organizational requirements, the methodology of testing, a list of maLerials to be delivered, and a schedule of user orientation.

d. To provide a written record of the actual test inputs to exercise system limits and critical capabilties, the instruction to permit execution of the test by the user staff and operator personnel, and the expected outputs.

1.2 Project References. This paragraph shall provide a brief summary of the references applicable to the history and development of the project. Documentation describing systems or procedures which supplement or provide for interaction with the subject system during the course of normal operation or at any point in the testing shall be specified. Included shall be a listing of at least all applicable documentation prepared for this project.

1.3 Terms and Abbreviations. This paragraph shall provide a listing or include in an appendix any terms, definitions, or acronyms unique to this document and subject to interpretation by the user of the document. This listing will not include item names or data codes.

SECTION 2. DEVELOPMENT TEST ACTIVITY

2.1 Statement of Pretest Activity. This paragraph shall provide a description of the testing completed as part of the system development activity. It may refer to an appendix in the form of a listing of the elements in the PS and SS against which the program operation has been explicity checked.

2.2 Pretest Activity Results. This paragraph will provide an overall evaluation of the test results obtained from all tests conducted as part of the system development activity. It may refer to an appendix when multiple tests have been conducted. Included must be a statement of any known system deficiencies and their potential impact.

SECTION 3. TEST PLAN

3.1 System Description. A brief description or system chart showing inputs and outputs shall be included to provide a frame of reference for the description of the tests to be conducted.

3.2 Testing Schedule. This paragraph shall provide a listing or chart depicting the locations at which the testing will be scheduled and the time frames during which the testing will be conducted.

3.3 First Location (Identify) Testing. This paragraph shall identify the first location at which the testing will be conducted and the participating organizations. This paragraph will also list the test(s) to be performed at this location and will reference the appropriate section(s) that describe the test(s).

3.3.1 Milestone Chart. This paragraph shall provide a chart to depict the activities and events listed below. An example of this type of chart is shown in Figure 3-01. When preparing this chart, consideration will be given to all tests scheduled for this location. The chart will be in chronological order with a supporting narrative as necessary with supporting narrative as necessary and will show, for example:

a. Overall on-site test period by calendar date and portions of the period assigned to major portions of test.

b. Pretest on-site period required for system debugging, orientation, and familiarization.

c. Period assigned for the collection of data base values, input values, and other operational data required for system test.

d. Period assigned for user orientation and familiarization with system documentation.

e. Period assigned for user training, operator training, maintenance and control group training, and orientation briefing for non-ADP personnel.

f. Period assigned for preparation, review, and approval of the Test Analysis Report.

3.3.2 Equipment Requirements. This paragraph shall provide a chart or listing to depict the period of usage and quantity required of each item of equipment employed throughout the test period at each location where the system is to be installed. Any test data reduction equipment shall be included.

3.3.3 Software. This paragraph shall list software programs, including systems support and applications programs, and their form (tape, disk, etc.) used during and in support of the testing when they are not a part of the system being tested.

3.3.4 Personnel. This paragraph shall provide a chart or a listing of the number and period of use of personnel with indicated skill types required during the entire test period(s). It will indicate special requirements, such as multishift operation and assignment or the retention of key skills to ensure continuity and consistency in extensive test programs. This chart or listing should be related to the milestone chart.

3.3.5 Orientation Plan. This paragraph will describe the number of personnel providing training during the testing and the types of training to be undertaken. This information shall be related to the personnel requirements in paragraph 3.3.4. This plan shall include user instruction (classroom and computer), operator instruction, maintenance and control group instruction (if applicable), and the orientation briefing of noncomputer-oriented staff personnel. (If extensive training is anticipated, a separate manual may be developed and referenced here.)

3.3.6 Test Materials. This paragraph shall itemize the articles and apparatus associated with conducting test.

3.3.6.1 Deliverable Materials. This paragraph will itemize all materials that will be delivered as part of the system to include the quantity and full identification. Examples of these include:

a. Users, Computer Operation, and Program Maintenance Manuals.

b. Program card decks/listings/card image tapes/binary tapes/etc.

c. Data base card decks/listings/card image tapes/etc.

d. Sample listings of system output.

3.3.6.2 Site Supplied Materials. All items that are expected to be provided shall be listed in this paragraph. Examples of these include:

a. Data base and its form card deck, tape, etc.), fully identified.

b. Other inputs and thier form.

c. Test control programs or other special test programs and their form.

d. Test worksheets and other forms or instructions prepared to control and expedite the test activity. Their type, layout, and quantity must be fully explained.

e. Apparatus required during or in support of test, when it is not normally part of the equipment configuration or is not being delivered as part of the installation effort. Examples of this apparatus are extra peripherals (tape drives, printers, plotters), test message genrators, test timing devices, test event records, etc. Such apparatus must be identified by name, type, and quantity required.

3.3.7 Security. Describe any security considerations associated with this test in addition to any privacy restrictions associated with the data being handled.

3.4 Second Location (Identify) Testing. This paragraph shall describe the testing to be conducted at the second location using the same information as outlined in paragraph 3.3.

```
                        Generate and   Selectively    Produce
     FUNCTION           maintain the   retrieve       special
                        data base      data           catalog
    (paragraph)         (3.2.1)   (3.2.2)        (3.2.3)

    Program short name            G D A P M F D D D P D N D
                                  N L S R D U A S A R I A P
                                  U P S O P A E P T N R M E
                                  P R C J R R L R A T T E R
Number/test
1.  Add new record            X
2.  Add new SORU                   X
3.  Add a card                     X
4.  Change a card                  X
5.  Delete a card                  X
6.  Delete a record                X
7.  Delete a SORU                  X
8.  Create index file                   X
Retrieve records with a requested:


11.  Functional area                         X
12.  Record ID code (RIC)                    X
13.  Several RICS                            X
14.  Index set and SORU                           X
15.  Index set                                    X
16.  2 different index sets                       X
17.  3 different index sets                       X
18.  RIC and associated records                   X
19.  Print full data base                              X
20.  Extract record directory                          X
21.  Extract record names/RICs                         X
22.  Extract keywords, permuted                             X
```

FIGURE 4-01. Test/Function Relationships for the Record association System

FIGURE 4-01. Example of Test/Function Relationship Chart

SECTION 4. TEST SPECIFICATION AND EVALUATION

4.1 Test Specification.

4.1.1 Requirements. This paragraph shall list the indvidual requirements to be demonstrated by the test as derived from the Specific Performance Requirements given in paragraph 3.1 of the FD.

4.1.2 System Functions. This paragraph shall provide a detailed list of the system functions which will be exercised during the overall system testing. This list, derived from paragraph 4.2 of the FD, must be ordered in such a way that the functions are related to the requirements given in paragraph 4.1.1.

4.1.3 Test/Function Relationships. This paragraph shall provide a list of the tests which, taken as a whole, constitute the overall test activity. It will also provide, as applicable, Test/Function matrix chart summarizing the overall allocation of the system functions to the test. Figure 4-01 shows an example of this type of chart.

4.2 Test Methods and Constraints.

4.2.1 System Test Conditions. This paragraph shall indicate whether the system test is to be made using the normal system input and data base or whether a special set of inputs and data base to be used.

4.2.2 Extent of System Test. This paragraph shall indicate the extent of the testing to be employed. Where total testing is not to be employed, the test requirements will be presented either as a percentage of some well defined total quantity or as a number of samples of discrete operating conditions or values. Also indicated will be the rationale for adopting limited testing.

4.2.3 Data Recording. This paragraph shall indicate data recording requirements, including those data types not normally recovered from system operation.

4.2.4 System Test Constraints. This paragraph shall indicate the anticipated limitations imposed on the test due to system or test conditions, such as limitations on timing, interfaces, equipment, personnel, and data base.

4.3 Test Progression. In cases of progressive or cumulative test, an explanation shall be included concerning the manner in which progression is made from one test to another so that the cycle or activity for each test is completely accomplished.

4.4 Test Evaluation

4.4.1 Test Data Criteria. This paragraph shall described the rules by which test results wll be evaluated; for example:

a. Tolerances - range over which a data value output by a system performance parameter can vary and still be considered acceptable.

b. Samples - the minimum number of combinations or alternatives of input conditions and output conditions that can be exercised to constitute an acceptable test of the parameters involved.

c. Counts - the maximum number of interrupts, halts, or other system breaks which may occur due to nontest conditions.

4.4.2 Test Data Reduction. This paragrph shall describe the technique to be used for manipulation of the raw test data into a from suitable for evaluation, if applicable. The available techniques could include:

a. Manual; i.e., manual collection and collation of system test outputs into test sequence order followed by visual inspection of the results.

b. Semiautomatic; i.e., automatic inspection of test results as obtained by data recording means using a test data reduction program followed by manual (visual) inspection of selected test reszults which do not lend themselves to complete reduction by automatic means.

c. Automatic; i.e., automatic inspecton of test results specifically recorded for manipulation by the test data reduction program. Test results, as recorded, include all items of test significance. The test date reduction program contains an image of correct data output for an item-by-item comparison of data and provides a summary of an evaluated test as output.

SECTION 5. TEST (IDENTIFY) DESCRIPTION

This section shall describe major logical groups of tests to be performed. If appropriate, additional logical groups shall be described in subsequent sections following the same format as that used in this section.

5.1 Test Description. This paragraph shall provide a general description of the test to be performed.

5.2 Test Control.

5.2.1 System Test Means of Control. This paragraph shall indicate whether the test is to be controlled by:

a. Manual means; i.e., manual inspection of necessary inputs and manual control of test sequence.

b. Semiautomatic means; i.e., manual insertion of necessary inputs and automatic (test program) control of test sequence.

c. Automatic means; i.e., preparation and use of a special test program to provide necessary input, conduct tests, monitor and record test results.

5.2.2 Test Data. In each of the folllowing paragarphs identify any security considerations.

5.2.2.1 Input Data. This paragraph will describe the manner in which input data are controlled in order to:

a. Test the system with a minimum number of data types and values.

b. Exercise the system with a range of bona fide data types and values which test for overload, saturation, and other "worst case" effects.

c. Exercise the system with bogus data types and values which test for rejection of irregular inputs.

5.2.2.2 Input Commands. This paragraph shall describe the manner in which input commands are used to control:

a. Initialization of test.

b. Halt or interrupt of test.

c. Repeat of unsuccessful or incomplete test.

d. Alternate modes of operation as required by test.

e. Termination of test.

5.2.2.3 Output Data. This paragraph shall describe the manner in which output data are analyzed in order to:

a. Detect occurrence (or ultimate nonoccurrence) of output data (as event.)

b. Record or identify media and location of all output data for indication of test performance.

c. Evaluate output as a basis for continuation of test sequence.

d. Evaluate test output against required output to assess the performance of the test.

5.2.2.4 Output Notification. This paragraph shall describe the manner in which output notifications (messages output by the system concerning status or limitations on internal performance) are controlled in order to:

a. Indicate readiness for test (normal operation condition).

b. Provide indications of irregularities in input test data or test data base due to normal or erroneous test procedures.

c. Provide indications of irregularities in internal operations on test data due to normal or erroneous test procedures.

d. Provide indications on the control, status, and results of the test as available from auxiliary test supervisor program (if used).

5.3 Test Procedures. This paragraph shall contain the step-by-step procedures to accomplish each test of the system. Each step shall be assigned a test step number and this number, along with critical test data and test procedures information, shall be tabulated onto a test procedure form for test control and the recording of test results.

5.3.1 Test Setup. If not stated elsewhere or by standard operating procedures, this paragraph shall itemize the activities associated with setup of the computer facilities to conduct the test, including all routine machine activities from "power on" through "console setup" to "card/tape read-in." Included shall be the distribution of test documents, worksheets, and other forms.

5.3.2 Test Initialization. This paragraph shall itemize in test sequence order the activities associated with establishing the conditions of the first test starting with the equipment in the setup condition. Initialization may include such functions as:

a. Readout of control function locations and critical data from indicators and storage locations for reference purposes.

b. Queuing of data input values for first test.

c. Queuing of test support programs, if used.

d. Coordination of personnel actions associated with test.

5.3.3 Test Steps. This paragraph shall itemize the test(s) into test steps in test sequence order. It shall also include special operations, such as:

a. Visual inspection of test conditions.

b. Data dumps.

c. Instructions for data recording.

d. Modifications of data base.

e. Interim evaluation of test results.

5.3.4 Test Termination. This paragraph shall itemize in test sequence order the activities associated with termination of the test, such as:

a. Readout of critical data from indicators and location for reference purposes.

b. Termination of operation of time-sensitive test-support programs and test apparatus.

c. Collection of system and operator records of test results.

4.17 Test Analysis Report. The RT (Test Analysis Report) describes the status of the computer program system after test and provides a presentation of deficiencies for review by staff and management personnel. Therefore, the document should be prepared in nontechnical language.

TEST ANALYSIS REPORT
TABLE OF CONTENTS

SECTION 1. GENERAL
- 1.1 Purpose of the Test Analysis Report
- 1.2 Project References
- 1.3 Terms and Abbreviations
- 1.4 Security

SECTION 2. TEST ANALYSIS
- 2.1 Test (Identify)
  - 2.1.1 System Function (Identify)
  - 2.1.2 Function Performance
  - 2.1.3 Parameter Performance
  - 2.1.4 Data Performance
- 2.2 Test (Identify)

SECTION 3. SUMMARY AND CONCLUSIONS
- 3.1 Demonstrated Capability
- 3.2 System Deficiencies
- 3.3 System Refinements

SECTION 1. GENERAL

1.1 Purpose of the Test Analysis Report. This paragraph shall describe the purpose of the RT (Test Analysis Report) in the following words or appropriate modifications thereto:

The Test Analysis Report For (Project Name) (Project Number) is written to fulfill the following objectives:

a. To document the results of the implementation test.

b. To provide a basis for allocationg responsibility for deficiency correct and follow-up.

c. To provide a basis for the preparation of the statement of project completion.

d. To establish user confidence in the operation of the system.

1.2 Project References. This paragraph shall provide a brief summary of the project objectives and identify the project sponsor and user. Also provided shall be a list of applicable documents by author or source, reference number, title, date, and security classification. This paragraph will include the folowing, when applicable:

b. Users Manual.

c. Computer Operation Manual.

d. Program Maintenance Manual.

e. Test Plan.

1.3 Terms and Abbreviations. This paragraph shall provide a listing or include in an appendix any terms, definitions, or acronyms unique to this document and subject to interpretation by the user of the document. This listing will not include item names or data codes.

1.4 Security. This paragraph will describe any security considerations associated with the test analysis and the data being handled.

SECTION 2. TEST ANALYSIS

2.1 Test (Identify). Each test shall be separately identified in paragraphs 2.1 through 2.n and must be related to those presented in the Test Plan.

2.1.1 System Function (Identify). Each system function, corresponding to the functions presented in paragraph 4.1.2 of the PT, shall be separately described.

2.1.2 Function Performance. This paragraph shall describe the capability as it has been demonstrated in one or more system tests. It shall also assess the manner in which the environment may be different from the operational environment and the effect of this difference on the capability.

2.1.3 Parameter Performance. This paragraph shall compare the parameter performance of the test with the parameter performance described in the UM, OM, and PT, when applicable.

2.1.4 Data Performance. This paragraph shall compare the I/O performance of the test with the I/O capabilities as described in the UM and the PT when applicable.

2.2 Test (Identify). This paragraph shall describe the second test in a manner similar to that used in paragraph 2.1.

SECTION 3. SUMMARY AND CONCLUSIONS

3.1 Demonstrated Capability. This paragraph shall provide a general statement of the capability of the system as demonstrated by the test, compared with the performance requirements contained in the system Functional Description. An individual discussion of conformance with specific requirements may be included on complex systems.

3.2 System Deficiencies. As required by the results of the testing, an individual statement will be provided for each deficiency in system operations, as measured against the Test Plan. Accompanying each deficiency will be a discussion of the impact:

a. On system performance if the deficiency is not corrected.

b. On the system design if the deficiency is corrected, along witn the assignment of organizational responsiblity for the correction.

3.3 System Refinements. An itemization of improvements which can be realized in system design or operation, as determine during the test period, will be given. Accompanying each improvement will be a discussion of the added capability it provides the system and the impact on the system design.

4.18 Implementation Pro
for directing the instal
than the test site after
be used to direct the im
an ADS which has already
toward the staff personn
those parts of the docum
presented in suitable te

edures. The IP (Implementation Procedures) is a tool ation or implementation of an ADS at locations other testing of the ADS has been completed. It may also lementation of major modifications or enhancements of been installed. Those parts of the document directed el shall be presented in nontechnical language and ent directed toward the operations personnel shall be minology.

IMPLEMENTED PROCEDURES
TABLE OF CONTENTS

SECTION 1. GENERAL
- 1.1 Purpose of the Implementation Procedures
- 1.2 Project References
- 1.3 Terms and Abbreviations

SECTION 2. IMPLEMENTATION OVERVIEW
- 2.1 Description
- 2.2 Contact Point
- 2.3 Support Materials
- 2.4 Training
- 2.5 Tasks
- 2.6 Personnel Orientation
- 2.7 Personnel Requirements
- 2.8 Security

SECTION 3. SITE INFORMATION - COMPUTER OPERATIONS
- 3.1 Site (Identify)
  - 3.1.1 Schedule
  - 3.1.2 Software Inventory
  - 3.1.3 Facilities
  - 3.1.4 Implementation Team
  - 3.1.5 Detailed Procedures
  - 3.1.6 Data Update Procedures
- 3.2 Site (Identify)

SECTION 4. SITE INFORMATION - USER
- 4.1 Site (Identify)
  - 4.1.1 Schedule
  - 4.1.2 Detailed Procedure
  - 4.1.3 Data Update Procedures
- 4.2 Site (Identify)

SECTION 1. GENERAL

1.1 Purpose of the Implementation Procedures. This paragraph shall describe the purpose of the IP (Implementation Procedures) in the following words, modified as appropriate:

The objective of the Implementation Procedures for (Project Name) (Project Number) is to provide the necessary information to the functional users and data processing personnel to accomplish the installation of a previously tested ADS and to achieve operational status at additional sites.

1.2 Project References. This paragraph shall provide a brief summary of the references applicable to the history, development, operation, and maintenance of the system. As a minimum, the following shall be specified by source or author, reference number, title, date, and security classification.

a. Project request.

b. Users Manual.

c. Computer Operation Manual.

d. Other pertinent documentaion on the project.

1.3 Terms and Abbreviations. This paragraph shall provide a list or include in an appendix any terms, definitions, or acronyms unique to this document and subject to interpretation by the user of this document. This list will not include item names or data codes.

SECTION 2. IMPLEMENTATION OVERVIEW

Section 2 shall provide a description of the implementation process, including support user, and operations activities. It shall also identify who will accomplish the various segments of the implementation process, provide a schedule of events, and present additional information of common interest to both the functional user and data processing personnel.

2.1 Description. This paragraph shall provide a general description of the implementation process to provide a frame of reference for the remainder of the document. A list of sites for ADS installation, the schedule dates, and the method of implementation shall be included.

2.2 Contact Point. This paragraph shall provide the organizational name, office symbol/code, and telephone number of a contact for questions relating to this implementation.

2.3 Support Materials. This paragraph shall list the type, source, and quantity of support materials required for the implementation. Included shall be items such as magnetic tapes, disk packs, punch cards, computer printer paper, and special forms.

2.4 Training. This paragraph shall describe the type and amount of special training required, if any.

2.5 Tasks. Each task required for the system installation shall be described or listed in general terms. Each task shall be identified with the organization that will accomplish this task, usually either the user, computer operations, or the developer. This task list or description shall include such items to be accomplished as:

a. Providing overall planning and coordination for the implementation and preparation of the Implementation Procedures.

b. Ensuring that all manuals applicable to the installation effort are available when needed.

c. Providing technical assistance.

d. Establishing criteria and supervising/conducting training activities associated with the implementation.

e. Scheduling processing required for the implementation.

f. Providing comprehensive support for the implementation.

g. Ensuring that all prerequisities have been fulfilled prior to the implementation date.

h. Providing personnel to the implementation team.

i. Arranging lodging, transportation, and office facilities for the implementation team.

j. Providing instructor and student personnel for training before and during the implementation effort.

k. Providing computer support.

l. Providing priority scheduling to ensure adequate turn around.

2.6 Personnel Orientation. This paragraph shall identify those efforts such as briefings and seminars intended to orient personnel to the new system.

2.7 Personnel Requirements. This paragraph shall describe the number, time, and skill level of the personnel required during the implementation period, including the need for multishift operation, clerical support, etc.

2.8 Security. This paragraph shall contain an overview and discussion of the security considerations associated with the data of the system.

SECTION 3. SITE INFORMATION - COMPUTER OPERATIONS

3.1 Site (Identify). The site or sites to be discussed in this paragraph shall be identified. Additional paragraphs (paragraphs 3.2 through 3.n) will be prepared as necessary to cover other sites. Multiple sites may be discussed within paragraph 3.1 when the information is generally consistent.

3.1.1 Schedule. This paragraph shall present a schedule of activities to be accomplished during implementation. It will depict the required tasks in chronological order with beginning and ending dates of each task with supporting narrative as necessary.

3.1.2 Software Inventory. This paragraph shall provide an inventory of the software required to support the implementation. The software will be identified by name, identification code or acronym, and security classification. It will be indicated if the software is expected to be on site or will be delivered for the implementation. Any software to be used only during the implemenation process shall be identified.

3.1.3 Facilities. This paragraph shall detail the physical facilities and accommodations required during the implementation period. Some of the factors to be considered are:

a. Classroom/work space/training aids needed; hours per day, number of days, and shifts.

b. Hardware that must be operational and available.

c. The availability of transportation and lodging for the implementation team.

3.1.4 Implementation Team. When an implementation team is required, this paragraph shall describe its composition. Each team member's task should be defined.

3.1.5 Detailed Procedures. Paragraphs 3.1.5.1 through 3.1.5.n shall provide in step-by-step sequence the detailed procedures required to accomplish the implementation. Reference may be made to other documents, i.e., Computer Operation Manual. Examples of the areas to be considered are:

a. Control inputs.

b. Operating instructions.

c. Input-output files.

d. Output reports.

e. Special handling.

f. Diagnostic messages.

g. Recovery/restart procedures.

3.1.6 Data Update Procedures. This paragraph shall present the data update requirements that may occur during the implementation period. When the data update procedures are the same as the normal updating/processing procedures, reference may be made to the OM. Paragraph 3.1.6.1 through 3.1.6.n shall provide step-by-step procedures to update the converted data. Examples of the areas to be considered are:

a. Control inputs.

b. Operating instruction.

c. Input-output files.

d. Output reports.

e. Special handling.

f. Diagnostic messages.

g. Restart/recovery procedures.

3.2 Site (Identify). This paragraph shall describe the computer operations at the second site using the same information as outlined in paragraph 3.1.

SECTION 4. SITE INFORMATION - USER

This section shall provide users with the information necessary to accomplish an orderly implementation. When more than one functional user is involved, a separate section (Sections 5 through N) may be written for each user and the section titles modified to reflect each user.

4.1 Site (Identify). The site or sites to be discussed in this paragraph shall be identified. Additional paragraphs (paragraphs 4.2 through 4.n) will be prepared as necessary to cover all sites. Multiple sites may be discussed within this paragraph when the information is generally consistent.

4.1.1 Schedule. This paragraph shall present a schedule of activities to be accomplished by the user during implementation. It will depict the required tasks in chronological order with beginning and ending dates of each task with supporting narrative as necessary.

4.1.2 Detailed Procedures. Paragrah 4.1.2.1 through 4.1.2.n shall provide in step-by-step sequence the detailed procedures required to accomplish the implementation. Reference may be made to other documents such as the UM. Examples of the areas to be considered are:

a. Initiation procedures.

b. Input formats.

c. Output formats.

d. Utilization of outputs.

e. Recovery and error correction procedures.

4.1.3 Data Update Procedures. This paragraph shall present the user's data update requirements that may occur during the implementation period. When update procedures are the same as normal processing, reference may be made to the UM and, if appropriate, to Section 3 of this document. Paragraphs 4.1.3.1 through 4.1.3.n shall provide step-by-step procedures for the update. Examples of areas to be considered are:

a. Initiation procedures.

b. Input formats.

c. Output formats.

d. Utilization of outputs.

e. Recovery and error correction procedures.

4.2 Site (Identify). This paragraph shall describe the user information at the second site using the same information as outlined in paragraph 4.1.

SECTION 3. PROJECT COMPLEXITY.

4.18. INTRODUCTION. As an aid to determining the amount of documentation required for a given project, a method has been developed which relates the complexity of a project to the amount of documentation required for the project. While the use of this method produces only general guidance, it does give the developer an idea of the document types that will be needed to satisfy the requirements for a project.

4.19. DETERMINING COMPLEXITY. In determining the complexity of a project, numerical values (1 - 5) are assigned to each of the factors shown in Figure III-01, and then adding them together to arrive at a complexity total. This total is then used to reference Figure III-02 to determine what document types may need to be written for the project. If, for example, the total were 14, either a Users' Manual; or a series of documents containing a Users' Manual, Computer Operations Manual, Program Maintenance Manual, and a Test Manual may be included.

Once the amount of documentation has been determined, the question of when must this documentation be produced must be answered. This can best be answered by relating the documents to phases and stages of the system's life, as shown in the following paragraphs:

a. Definition. A Functional Description is normally the only document produced during this stage. If the situation calls for the development of a Data Requirements Document, it may also be produced during this stage, but it is normally completed after the Functional Description.

b. Design. If this stage is necessary, a System/Subsystem Specification, Program Specification, and (if necessary) a Data Base Specification may be produced.

c. Development and Integration. During this stage the final documentation in the form of a Users Manual, Computer Operation Manual, and Program Maintenance Manual may be prepared.

d. Test. A Test Plan may be prepared in this stage.

e. Evaluation. The Test Analysis Report (if necessary) is prepared during this stage.

f. Deployment and Operation. In this phase the Implementation Procedures may be produced.

Figure 4-12 indicates the normal period of time each document is used during the Automation Life Cycle. In actual practice, the sequence of preparation of the various documents is rarely as sharply defined as shown in Figure 4-12.

FACTORS / COMPLEXITY:	1	2	3	4	5
1. ORIGINALITY REQUIRED	NONE; REPROGRAM ON DIFFERENT EQUIPMENT	MINIMUM; MORE STRINGENT REQUIREMENTS	LIMITED; MORE ENVIRONMENT, NEW INTERFACES	CONSIDERABLE, APPLY EXISTING STATE OF ART TO ENVIRONMENT	EXTENSIVE; REQUIRES ADVANCE IN STATE OF THE ART
2. DEGREE OF GENERALITY	HIGHLY RESTRICTED SINGLE PURPOSE	RESTRICTED; PARAMETERIZED FOR A RANGE OF CAPACITIES	LIMITED FLEXIBILITY; ALLOWS SOME CHANGE IN FORMAT	MULTI-PURPOSE; FLEXIBLE FORMAT, RANGE OF SUBJECTS	VERY FLEXIBLE ABLE TO HANDLE A BROAD RANGE OF SUBJECT MATTER ON DIFFERENT EQUIPMENT
3. SPAN OF OPERATION	LOCAL OR UTILITY	COMPONENT COMMAND	SINGLE COMMAND	MULTI-COMMAND	DEFENSE DEPARTMENT, WORLD-WIDE
4. CHANGE IN SCOPE AND OBJECTIVE	NONE	INFREQUENT	OCCASIONAL	FREQUENT	CONTINUOUS
5. EQUIPMENT COMPLEXITY	SINGLE MACHINE ROUTINE PROCESSING	SINGLE MACHINE, ROUTINE PROCESSING, EXTENDED PERIPHERAL SYSTEM	MULTI-COMPUTER, STANDARD PERIPHERAL SYSTEM	MULTI-COMPUTER, ADVANCED PROGRAMMING COMPLEX PERIPHERAL SYSTEM	MASTER CONTROL SYSTEM, MULTI-COMPUTER, AUTO INPUT-OUTPUT AND DISPLAY EQUIPMENT
6. PERSONNEL ASSIGNED	1-2	3-5	5-10	10-18	18 AND OVER
7. DEVELOPMENTAL COST	1-10 K	10-50 K	50-200 K	200-500 K	OVER 500 K
8. CRITICALITY	DATA PROCESSING	ROUTINE OPERATIONS	PERSONNEL SAFETY	UNIT SURVIVAL	NATIONAL DEFENSE
9. AVERAGE RESPONSE TIME TO PROGRAM CHANGES	2 OR MORE WEEKS	1 - 2 WEEKS	3 - 7 DAYS	1 - 3 DAYS	1 - 24 HOURS
10. AVERAGE RESPONSE TIME TO DATA INPUTS	2 OR MORE WEEKS	1 - 2 WEEKS	1 - 7 DAYS	1 - 24 HOURS	0 - 60 MINUTES
11. PROGRAMMING LANGUAGES	HIGH LEVEL	HIGH LEVEL AND LIMITED ASSEMBLY LANGUAGE	HIGH LEVEL AND EXTENSIVE ASSEMBLY LANGUAGE	ASSEMBLY LANGUAGE	MACHINE LANGUAGE
12. CONCURRENT SOFTWARE DEVELOPMENT	NONE	LIMITED	MODERATE	EXTENSIVE	EXHAUSTIVE
TOTALS	X1=	X2=	X3=	X4=	X5=
					COMPLEXITY TOTAL:

FIGURE 4-12. Level of Project Complexity

COMPLEXITY TOTAL	DOCUMENT TYPES			
12 - 15			UM	
12 - 26			UM OM MM	PT
24 - 38	FD		UM OM MM	PT
36 - 50	FD	SS	UM OM MM	PT PT
48 - 60	FD	SS PS	UM OM MM	PT RT

NOTES:

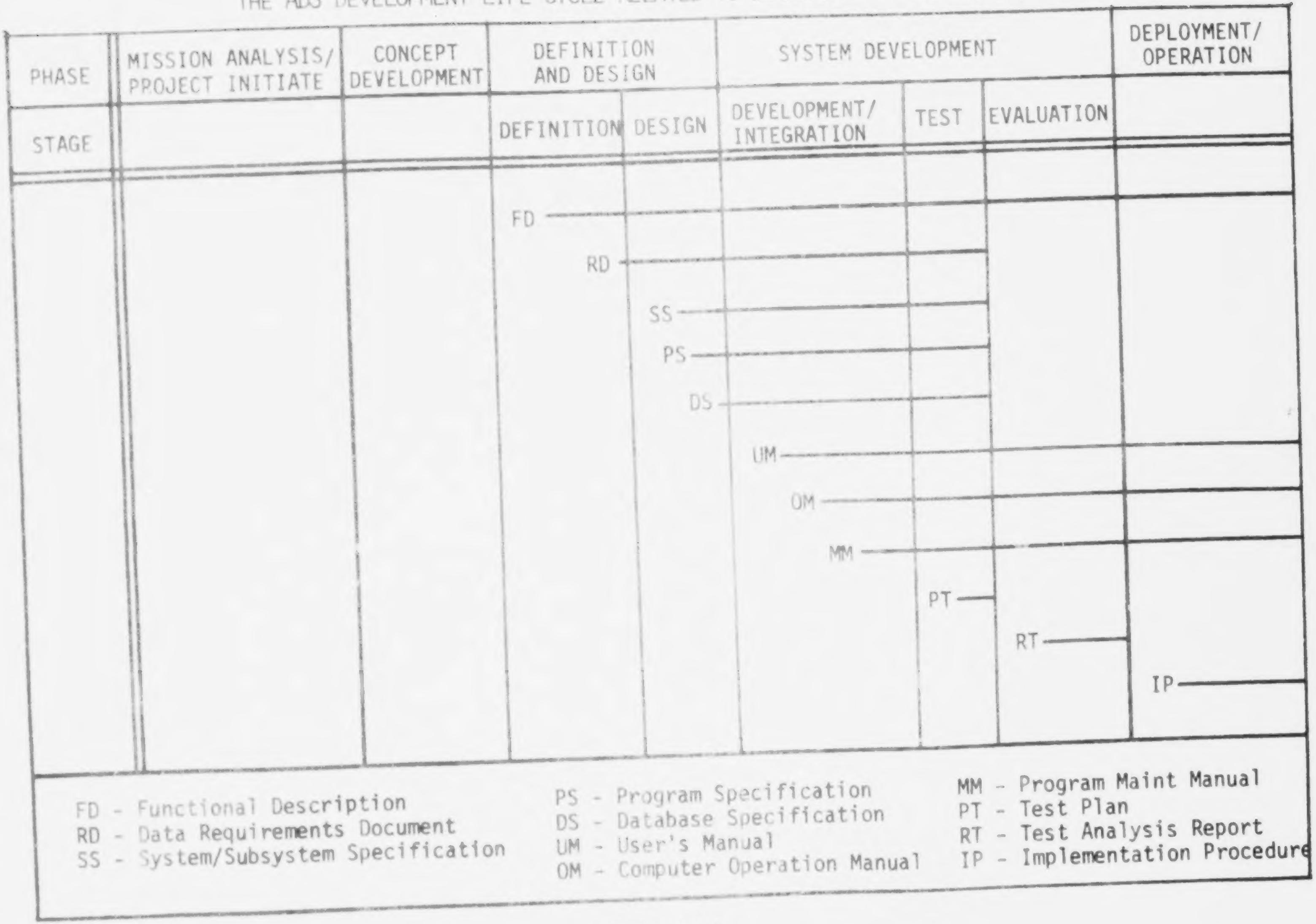
1. Preparation of the Data Requirements Document and the Data Base Specification is dependent upon the situation.
2. Additional document types may be required at lower complexity levels

ABBREVIATIONS:

FD - FUNCTIONAL DESCRIPTION
SS - SYSTEM/SUBSYSTEM SPECIFICATION
PS - PROGRAM SPECIFICATION
UM - USERS' MANUAL
OM - COMPUTER OPERATION MANUAL
MM - PROGRAM MAINTENANCE MANUAL
PT - TEST PLAN
RT - TEST ANALYSIS REPORT

FIGURE 4-13. Document Types/Project Complexity

THE ADS DEVELOPMENT LIFE CYCLE RELATED TO DOCUMENT PREPARATION AND USE



FD - Functional Description
RD - Data Requirements Document
SS - System/Subsystem Specification
PS - Program Specification
DS - Database Specification
UM - User's Manual
OM - Computer Operation Manual
MM - Program Maint Manual
PT - Test Plan
RT - Test Analysis Report
IP - Implementation Procedure

FIGURE 4-14.

AUTOMATION LIFE CYCLE

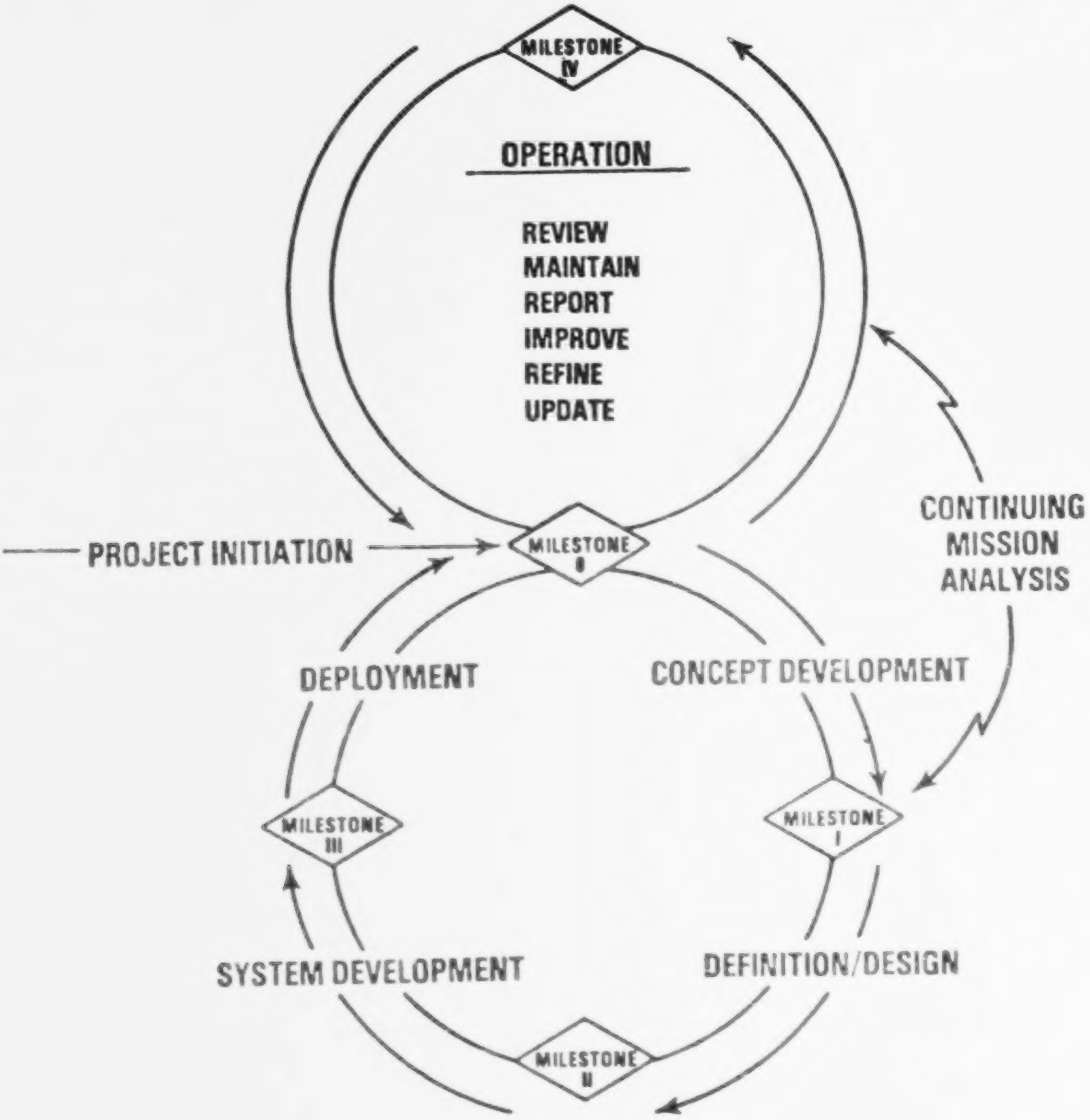

MILESTONE IV
OPERATION
REVIEW
MAINTAIN
REPORT
IMPROVE
REFINE
UPDATE
CONTINUING MISSION ANALYSIS
PROJECT INITIATION
MILESTONE 0
DEPLOYMENT
CONCEPT DEVELOPMENT
MILESTONE III
MILESTONE I
SYSTEM DEVELOPMENT
DEFINITION/DESIGN
MILESTONE II

FIGURE 4-14a.

SECTION 4. APPLICABLE DA FORMS.

4-20. INTRODUCTION. The following pages identify the Department of the Army Forms which are applicable to Systems Documentation and identify in which types of documentation the various forms are used. Instruction for the preparation of DA forms described in TB 18-111 is also to be found in this section.

4.21. LIST OF DA-APPROVED FORMS

FORM NUMBER	TITLE	APPLICATION
4735 -R	Logic Chart	Presents information graphically, such as data flow diagram, logic flow diagram, and interface diagram.
4736 -R	Decision Table	Presents information as a problem/solution diagram; processing decision; presenting alternatives or exceptions; processing sequences.
4737 -R	File Specification	Documents input, output, and intermediate files.
4738 -R	Record Specification	Records format layouts.
4739 -R	Input-Process-Output	Provides input-process-output (IPO) information about structured programming modules.
4740 -R	Input-Process-Output Cross Reference	Provides cross-reference information between IPO and IPO extended Description Forms.
4741 -R	Input-Process-Output Extended Description Process Number	Provides detailed information about selected items on IPO forms.
4742 -R	Cycle Summary	Summarizes information about each cycle within the ADS.
4743 -R	Control Card Preparation	Provides control card format and preparation instructions.
4743-1 -R	Control Card Preparation (Continuation)	Used as a continuation of the Control Card Preparation form.
4744 -R	Run Set-up Console Operating Information	Provides DPI operations personnel with run setup and console operating information.

FORM NUMBER	TITLE	APPLICATION
4745-R	Run Messages and Responses	Records program display messages and corresponding operator action required.
4748-R	Output Reports	Provides means of identifying printed listings and cards that are produced by each run.
4749-R	Reproduced Output Reports	Identifies computer-generated reports that are produced after a run is completed.
4750-R	PCM Processing Instructions	Provides punched-card machine (PCM) processing information.
4751-R	Recovery Instructions	Provides restart and recovery procedures required by operations personnel.
4752-R	Program Revision	Provides a history of changes to specific programs subsequent to initial program validation tests.

APPLICABILITY OF DA FORMS

DOCUMENT TYPES DA FORMS	FD	RD	SS	PS	DS	UM	OM	MM	PT	RT
3162-R	X	X	X	X	X	X	X	X	X	X
4486-R	-	-	-	-	-	-	-	-	X	X
4735	X	X	X	X	X	X	X	X	X	X
4736	X	X	X	X	X	X	X	X	X	X
4737	-	-	X	X	-	-	-	X	-	-
4738	-	-	X	X	-	-	-	X	-	-
4739	-	-	X	X	-	-	-	-	-	-
4740	-	-	X	X	-	-	-	-	-	-
4741	-	-	X	X	-	-	-	-	-	-
4742	-	-	-	-	-	-	X	-	-	-
4743	-	-	-	-	-	X	X	-	-	-
4743-1	-	-	-	-	-	X	X	-	-	-
4744	-	-	-	-	-	-	X	-	-	-
4745	-	-	-	-	-	-	X	-	-	-
4748	-	-	-	-	-	-	X	-	-	-
4749	-	-	-	-	-	-	X	-	-	-
4750	-	-	-	-	-	-	X	-	-	-
4751	-	-	-	-	-	-	X	-	-	-
4752	-	-	-	-	-	-	-	X	-	-

Figure 4-15

DA DOCUMENTATION FORMS PREPARATION INSTRUCTIONS

4.22. Preparation instructions for DA Forms 4735-R through 4745-R and 4748-R through 4752-R are published here. Most entries on the forms in this appendix are completed by the originator of the form. The preparation instructions follow a concept of the originator completing the form when the information is known. Unusual conditions lead to the phrases, "DPI entry" and "DPI may modify this entry", when it is recognized that subsequent users of the form will possess additional or updated information. These instructions have been extracted from TB 18-111 to provide guidance in instances where the TB is not available.

DA FORM 4735-R
PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) chart was initially prepared or revised.
BLOCK 2. ID. Enter the appropriate system, cycle, run, or program identification code.
BLOCK 3. TITLE. Enter the appropriate descriptive title of the system, cycle, run, or program.

DA FORM 4736-R—DECISION TABLE
PREPARATION INSTRUCTIONS

BLOCK 1. TABLE TYPE. Check appropriate space to denote if the table type is open or closed.
BLOCK 2. DATE. Enter date (YYMMDD) this form was prepared or revised.
BLOCK 3. ID. Enter the identification code assigned to the system and/or program described in this table. (See TB 18-103).
BLOCK 4. TITLE. Enter the descriptive title and unique number assigned to the decision table.
COLUMN 5. SEQ NO. Enter identifier number for each condition/action identified in COLUMN 6. Numbers should be in numeric sequence, starting with "01".
COLUMN 6. CONDITIONS/ACTIONS. Upper part of table should be used as the Condition Stub Part; lower part used as the Action Stub Part.

a. CONDITION STUB PART. The condition stub part of the condition area is used to express questions or make statements concerning the problem being analyzed.

b. ACTION STUB PART. The action stub part of the action area is used to express commands or indicate actions to be taken on or with data when performing an operation.

COLUMN 7. RULES. Upper part of table should be used as the Condition Rule Part and the lower part used as the Action Rule Part.

a. CONDITION RULE PART. The condition rule part of the condition area is used to indicate the relationship of a particular condition stub entry to an action stub entry.

b. ACTION RULE PART. The action rule part of the action area is used to indicate the relationship of a particular action stub entry to a condition entry or combination of entries.

DA FORM 4737-R—FILE SPECIFICATION
PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) this file specification was prepared or revised.
BLOCK 2. ID. Enter the identification code assigned to the file. (See TB 18-103).
BLOCK 3. TITLE. Enter the descriptive title assigned to the file, such as "Stock Control File".
BLOCK 4. FILE DESCRIPTION. Enter a brief narrative description of the file.
BLOCK 5. STORAGE MEDIA. Enter the device ("tape", "disk pack", "drum", etc.) on which the file will be stored.
BLOCK 6. RECORD FORMAT. Enter "fixed", "variable", or "undefined"; see vendor's manual for other allowable formats.
BLOCK 7. FILE ORGANIZATION. Enter file organization, such as "sequential", "random", or "indexed sequential".

BLOCK 8. SECLAS/PRIV. Enter the security classification code (e.g., U = unclassified, C = co dential (AR 18–12–10)) and the privacy considerations (Y = yes, N = no) for the file.

BLOCK 9. TAPE REC MODE. Enter the mode in which the data is recorded, such as "BCD" (Bin Coded Decimal) or "EBCDIC" (Extended Binary Coded Decimal Interchange Code).

BLOCK 10. LABEL FORMAT. Indicate label conventions, such as "standard", "omitted", or "r standard".

BLOCK 11. SEQUENCE. Enter the primary and subordinate keys used to sequence records on a such as "day within month", "within year" for chronological sequence. Keys may include record posit or reference to a data dictionary.

BLOCK 12. PARITY. (For tape only). Enter type of parity ("odd" or "even").

BLOCK 13. TAPE BPI/TRACK. Enter recording density ("800", "1600", or "6250 BPI") and nun of tracks ("7" or "9").

BLOCK 14. MIN-MAX SPACE REQ. Estimate file size showing anticipated minimum and maxir number of records. Estimate growth factor, if known.

BLOCK 15. BLOCK FACTOR. (For tape only). Enter number of logical records in a physical rec

BLOCK 16. RETENTION. Enter the length of time the file will be retained. Express in term number of versions, cycles, or other period of time due to the occurrence of an event, e.g., "upon fication of final report." DPI may modify entries.

a. UPON TERMINATION. Entry should reflect planned retention of the last "current" file termination of the system or use of the file. If a precise time period cannot be determined, enter: "WH DET NLN FOR MSN RQMTS" which is the abbreviation for "When determined no longer neces for mission requirements."

b. UPON UPDATE. Enter the length of time the "prior" file is to be retained, i.e., the "current once superseded by cyclic update or other replacement version.

c. DUPLICATE(S). If applicable, this entry should reflect the length of time for retention of auxi transactions or duplicate files, which are retained for historical, security backup, or other similar poses. Not to be included are duplicate files which are prelocated as a part of the HQDA COOP.

BLOCK 17. RECORD TITLE. List the descriptive name for each type of record to be maintaine the specified file, e.g., "Material Release Order" or "Material Receipt Confirmation". Record title be identical to that shown on DA Form 4738–R (Record Specification).

COLUMN 18. RECORD ID. Enter the unique numeric code assigned to each record in the sy A numeric code will be assigned by the developer to each record utilized in the system. This will p the Record Specification forms (DA Form 4738–R) to be arranged in numeric sequence for ease of s and provide unique identification in addition to the title of the record.

COLUMN 19. % FILE. Enter estimate of the percentage of the overall file represented by the re

BLOCK 20. REMARKS. Enter any other pertinent information.

DA FORM 4738–R—RECORD SPECIFICATION
PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) this form was prepared or revised.

BLOCK 2. ID. Enter the Record ID as recorded in Column 18 of DA Form 4737–R.

BLOCK 3. TITLE. Record the descriptive title of the record.

BLOCK 4. DESCRIPTION. Provide a general description of the purpose of the record. Do not refe specific files or programs.

BLOCK 5. LENGTH. Record the record length in number of characters without regard to reco mode. For variable length, describe min-max length.

BLOCK 6. SECLAS/PRIV. Record the security classification code (e.g., U = unclassified, C = dential (AR 18–12–10)) and the privacy considerations (Y = yes, N = no).

COLUMN 7. POSITION. Provide the starting and ending position numbers for the field. If th is only one position long, only one number will be entered.

COLUMN 8. FIELD. Consecutively number each record field.

COLUMN 9. FIELD TITLES. Provide the descriptive title of the field. Include standard data ele data use identifiers, if possible.

COLUMN 10. CLASS. Record either A for Alphabetic, AN for Alphanumeric, N for Numeric, Special Characters, or NS for Numeric Signed, that describes the field.

COLUMN 11. LENGTH. Enter the number of characters.

COLUMN 12. REMARKS. Enter any appropriate additional information such as file key, sequence information, or information necessary to validate or otherwise process the field.

DA FORM 4739–R—INPUT-PROCESS-OUTPUT PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) this form was prepared or revised.
BLOCK 2. MODULE ID. Enter the identification code assigned to the module. (See TB 18–103).
BLOCK 3. MODULE NAME. Enter the descriptive name assigned to the module.
BLOCK 4. VERSION NO. Enter the latest version number of the module described on this form.
BLOCK 5. ANALYST. Enter the name, office symbol and telephone number of the analyst assigned to this module.
BLOCK 6. PROGRAMER. Enter the name, office symbol and telephone number of the programer assigned to this module.
BLOCK 7. INPUT. Identify, in numeric sequence and by name, each input provided to the module.
BLOCK 8. PROCESS. Identify, in numeric sequence and by name, each processing subfunction performed by the module.
BLOCK 9. OUTPUT. Identify, in numeric sequence and by name, each output produced by the processing subfunctions of the module.

DA FORM 4740–R—INPUT-PROCESS-OUTPUT CROSS REFERENCE PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) this form was prepared or revised.
BLOCK 2. MODULE ID. Enter the identification code assigned to the module (See TB 18–103).
BLOCK 3. MODULE NAME. Enter the descriptive name assigned to the module.
BLOCK 4. VERSION NO. Enter the latest version number of the module described on this form.
BLOCK 5. ANALYST. Enter the name, office symbol and telephone number of the analyst assigned to the module.
BLOCK 6. PROGRAMER. Enter the name, office symbol and telephone number of the programer assigned to the module.
COLUMN 7. IPO NUMBER. Enter, in the appropriate column, each numbered Input, Process, Output (from DA Form 4739–R) that requires an extended description on DA Form 4741–R.
COLUMN 8. EXTENDED DESCRIPTION ITEMS. For each IPO number entered in Column 7, enter the corresponding title of the associated Input, Process, and Output as contained on DA Form 4739–R.
COLUMN 9. PAGE REF. Reference page numbers of DA Form 4741–R on which the extended description items identified in COLUMN 8 are described.

DA FORM 4741–R—INPUT-PROCESS-OUTPUT EXTENDED DESCRIPTION, PROCESS NUMBER PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) this form was prepared or revised.
BLOCK 2. MODULE ID. Enter identification code assigned to the module. (See TB 18–103).
BLOCK 3. MODULE NAME. Enter the descriptive name assigned to the module.
BLOCK 4. VERSION NO. Enter the latest version number of the module described in the form.
BLOCK 5. ANALYST. Enter the name, office symbol and telephone number of the analyst assigned to the module.
BLOCK 6. PROGRAMER. Enter the name, office symbol and telephone number of the programer assigned to the module.
BLOCK 7. Provide the detailed information necessary to explain the Input, Process or Output to be accomplished.

DA FORM 4742–R—CYCLE SUMMARY PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) this form was prepared or revised.
BLOCK 2. CYCLE ID. Enter the identification code assigned to the cycle. (See TB 18–103).
BLOCK 3. CYCLE TITLE. Enter the descriptive title assigned to the cycle; e.g., "Daily Cycle", "As Required Cycle".
BLOCK 4. SUMMARY. The formatting of data in this part of the form may vary, depending upon the amount of detail required to adequately describe the cycle.
DA Form 4742–R is prepared in five parts; Part I through Part V. "Part I. Cycle Summary", contains

a brief narrative summary for each cycle. The cycles were originally depicted in paragraph 2.2, Sys Organization. Also include the following tabulation:

Run ID	*Run Title*
List the runs in the sequence they will run.	Provide the title of the run

The next entry on DA Form 4742–R is "Part II. Security and Privacy". In subparagraph "a. Secur list each classified component of the cycle and its classification. If no component is classified, so st "NA" is not an appropriate entry. In subparagraph "b. Privacy." list each component of the cycle requires safeguards for privacy and describe the privacy requirements. If no component is subje privacy considerations, so state; "NA" is not an appropriate entry.

The next entry on DA Form 4742–R is "Part III. Restrictions." Describe any restrictions associ with processing the cycle. Also list all approved waivers for the ADS that authorize exemptions processing standards.

The next entry on DA Form 4742–R is "Part IV. Interface." Complete the following tabulation:

File ID	*I/O*	*File Name*	*Source/Disposition*
List those files that are received from or are input to another cycle. Sequence by File ID	Record either "I", "O" or "I/O" with the associated file	Record the name of the associated file.	Record the Cycle ID of the source or the disposi the file. For files that a show the source cycle fo by the disposition cycle. "All cycles" is a valid For those files used in than one cycle "Library acceptable entry. Whe source of an input file ternal to the DPA reco means of receipt, i.e., DIN, hand-carry, etc. entry.

The next entry on DA Form 4742–R is "Part V. Other Information." Record subparagraph "a. Rep If no reports are produced in this cycle record just "NA". *Note:* "produced" is used herein to in prepared or made ready, but not necessarily printed. If reports are produced then prepare and com the following tabulation.

Report PCN and RCS	*Report Title*
List the PCN with the RCS (if available) to the right of PCN. Sequence by PCN in alphabetical numerical sequence	Record the title of the report.

The next subparagraph is "b. Control Cards." List by Control Card ID sequence each control car in the cycle. If none record "NA". The final subparagraph is "c. General Information." Recor general information that is not recorded elsewhere in this ADSM and is necessary to process this

DA FORM 4743–R—CONTROL CARD PREPARATION AND DA FORM 4743–1–R—CONTROL CARD PREPARATION (CONTINUATION) PREPARATION INSTRUCTIONS

BLOCK 1. **DATE.** Enter date (YYMMDD) this form was prepared or revised.

BLOCK 2. **RUN ID.** Enter the identification code assigned to the run. (See TB 18–103).

BLOCK 3. **RUN TITLE.** Enter the title of the run.

BLOCK 4. **CONTROL CARD ID.** Enter the identification code assigned to the control card. (S 18–103).

BLOCK 5. **CONTROL CARD TITLE.** Enter the title of the control card.

BLOCK 6. **CONTROL CARD PURPOSE AND REMARKS.** Define the purpose of the control ca

BLOCK 7. **USER INFORMATION.** If responsibility for information in this control card is assig the functional user, include such information as Authorized Agent, Frequency and Due Date Medium for Transmittal of Functional Data.

BLOCK 8. **CONTROL CARD INSTRUCTIONS.** Positions 1–13 must contain Control Card ID BLOCK 4 above.

COLUMN a—FIELD.—Enter the data name of the card field being defined.

COLUMN b—POSITION.—Enter the first and last column for the field specified in COLUMN a. All columns in the card will be recorded.

COLUMN c—PREPARATION INSTRUCTIONS.—Enter the necessary information to allow proper completion of the field in the control card. This entry may be a prescribed code, a list of optional entries, or a reference to a code table.

COLUMN d—RESPONSIBLE AGENCY.—Enter the agency responsible for specifying the field entry.

DA FORM 4744–R—RUN SETUP-CONSOLE OPERATING INFORMATION PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) this form was prepared or revised.

BLOCK 2. RUN ID. Enter identification code assigned to the run. (See TB 18–103).

BLOCK 3. RUN TITLE. Enter the title of the run.

BLOCK 4. SECLAS/PRIV. Enter the security classification code (e.g., U = unclassified, C = confidential (AR 18–12–10)) and the privacy considerations (Y = yes, N = no) of the run.

BLOCK 5. PERIPHERAL EQUIPMENT. Enter the type and number of peripheral equipment required for the run. Additional required peripheral equipment, equipment type, and the required number are to be entered on the undefined lines (OTHER).

BLOCK 6. CORE REQ. Enter the amount of core required to process the run.

BLOCK 7. POINT OF CONTACT. Identify local analyst and how to make contact if problems are encountered during processing.

BLOCK 8. RUN AS OF. Enter the "run as of" date.

BLOCK 9. RUN DATE. Enter the date that the run is to be made.

BLOCK 10. CONT CARD ID. For a single control card enter the assigned identification code for the control card and the ID Code (from Column 17) of the processing step where it is used. For multiple control cards enter "Yes—see Block 28" and in Block 28, enter the assigned identification code for each control card and the ID Code (from Column 17) of the processing step where it is used.

BLOCK 11. DATE/TIME SCH IN. Enter the due in date and/or time requirements. DPI entry.

BLOCK 12. DATE/TIME SCH OUT. Enter the due out date and/or time requirements. DPI entry.

BLOCK 13. SYSTEM. Enter the appropriate hardware system and/or the operating system.

BLOCK 14. RUN NO. Enter the run number, if appropriate. DPI entry.

BLOCK 15. EST. TIME. Enter the estimated wall clock or processing time in minutes.

BLOCK 16. LANGUAGE. Enter program language(s) used.

COLUMN 17. ID CODE. Enter identification code of the run subset (i.e., program, job step, PROC step) to be executed during the run. Sequence entries in processing order.

COLUMN 18. FILE ID. List the file ID or Volume Name (of multifile storage media) of the files used to process the corresponding run subset.

COLUMN 19. VOLUME NO. List the volume/reel number of the corresponding file.

COLUMN 20. I/O. Enter appropriate I/O code that represents the usage of the corresponding file. Use "I" for input, "O" for output, and "I/O" for input and output files.

COLUMN 21. MEDIA. Enter the media of the file. Use "LIST", "CARD", "TAPE", or "DISK", as applicable.

COLUMN 22. UNIT. Enter the physical unit assignment, type or number that corresponds to the location of the file identified in Column 18. When dynamically assigned this is a DPI entry.

COLUMN 23. RETN. Enter the retention period of the output file entered in Column 18. Entry may be in days (e.g., "DO5" = 5 days) or cycles (e.g., "CO5" = 5 cycles).

COLUMN 24. SECLAS/PRIV. Enter the security classification code (e.g., U = unclassified, C = confidential (AR 18–12–10)) and the privacy considerations (Y = yes, N = no) for each file.

COLUMN 25. SOURCE. Enter source from which the corresponding file entered in Column 18 is obtained; i.e., "SCRATCH", "BLANK CARDS", "LIBRARY", "DATA CONTROL", "ON-LINE", or another subset within the run. If spooling technique is used, enter "TAPE" or "DISK".

COLUMN 26. DISPOSITION. Enter appropriate disposition of the file entered in Column 18. For example, if the file is used in another subset within this run, enter the appropriate subset ID. If the file is used in another run within this cycle, enter the appropriate run ID. If the file is to be retained, enter "LIBRARY" the last time the file is used in the cycle. Files that remain on line will be entered as "ON-LINE". If the file requires off-site storage enter "OFF-SITE".

COLUMN 27. OPERATOR NOTES. Entries in this column are for operator use only.
COLUMN 28. REMARKS. This block is used to explain or clarify any entry on this form.

DA FORM 4745–R—RUN MESSAGES AND RESPONSES
PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) this form was prepared or revised.
BLOCK 2. RUN ID. Enter the identification code assigned to the run. (See TB 18–103).
BLOCK 3. RUN TITLE. Enter the descriptive title assigned to the computer run.
COLUMN 4. MESSAGE DISPLAYED. Enter the complete message displayed that requires operator action during the computer run.
COLUMN 5. ACTION REQUIRED. Identify appropriate action required in response to the displayed message. Entry such as "Notify Analyst" or "See Recovery Instructions" may be inserted, if appropriate.
BLOCK 6. REMARKS. Enter any additional information.

DA FORM 4748–R—OUTPUT REPORTS
PREPARATION INSTRUCTIONS

BLOCK 1. RUN ID. Enter the identification code assigned to the run. (See TB 18–103).
BLOCK 2. DATE. Enter the date (YYMMDD) this form was prepared or revised. DPI entry.
COLUMN 3. FILE ID. Enter the identification code assigned to each file. (See TB 18–103).
COLUMN 4. MEDIA. Enter the file media, i.e., tape, disk, card, etc.
COLUMN 5. PCN/RCS. Enter the Product Control Number (PCN) or the Requirement Control Symbol (RCS) for each report. (See TB 18–103 for PCN coding).
COLUMN 6. SECLAS/PRIV. Enter the security classification code (e.g., U = unclassified, C = confidential (AR 18–12–10)) and the privacy considerations (Y = yes, N = no) of each PCN/RCS on file.
COLUMN 7. PRINTER. These instructions are for the printing of reports.
COLUMN 7a. PAPER. Enter the form number or type of form.
COLUMN 7b. NO. OF CYS. Enter the number of copies required. DPI may modify this entry.
COLUMN 7c. TAPE. Enter the number of the carriage control tape or standard (STD).
COLUMN 7d. PRINT POS. Enter the printing positions.
COLUMN 7e. LINE UP. Enter the number of the first line to be printed.
COLUMN 8. VOLUME. Enter the number of forms or lines to be printed. Count only the first copy on multicopy reports. DPI entry.
COLUMN 9. REPORT TITLE. Enter the Report Title.
COLUMN 10. DISPOSITION. Enter the disposition of the printed reports.
BLOCK 11. REMARKS. Enter any clarifying information for the other entries.

DA FORM 4749–R—REPRODUCED OUTPUT REPORTS
PREPARATION INSTRUCTIONS

BLOCK 1. RUN ID. Enter the identification code assigned to the run. (See TB 18–103).
BLOCK 2. DATE. Enter the date (YYMMDD) this form was prepared or revised. DPI entry.
COLUMN 3. PCN/RCS. Enter the Product Control Number (PCN) or the Requirement Control Symbol (RCS) for each report. (See TB 18–103 for PCN coding).
COLUMN 4. SECLAS/PRIV. Enter the security classification code (e.g., U = unclassified, C = confidential (AR 18–12–10)) and the privacy considerations (Y = yes, N = no) of each PCN/RCS.
COLUMN 5. OUTPUT SIZE. Enter the size (face dimension) of each PCN/RCS report.
COLUMN 6. MEDIA. Enter the media, i.e., microfiche, paper, etc.
COLUMN 7. NO. OF CYS. Enter the number of copies required.
COLUMN 8. INSTRUCTIONS FOR REPORT PRODUCTION. Enter applicable instruction for the preparation of the PCN/RCS.
COLUMN 9. DISPOSITION. Enter the disposition of each PCN/RCS.
BLOCK 10. REMARKS. Enter only clarifying information for the other entries.

DA FORM 4750–R—PCM PROCESSING INSTRUCTIONS
PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter the date (YYMMDD) this form was prepared or revised. DPI entry.
BLOCK 2. RUN ID. Enter the identification code assigned to the run. (See TB 18–103).
BLOCK 3. RUN TITLE. Enter the title of the run.

BLOCK 4. PCN/RCS. Enter the appropriate Product Control Number (PCN) and/or Requirement Control Symbol (RCS), if applicable. (See TB 18-103 for PCN coding).

BLOCK 5. SOURCE. Enter the run subset ID from which the input is received.

BLOCK 6. DISPOSITION. Enter disposition (e.g., "DATA CONTROL") of processed cards.

BLOCK 7. SECLAS/PRIV. Enter the security classification code (e.g., U = unclassified, C = confidential (AR 18-12-10)) and the privacy considerations (Y = yes, N = no) of the PCN/RCS, if applicable.

COLUMN 8. STEP. List the steps required to complete the processing. Start with Step "01" and use the next sequential number for the next unique process, and so forth.

COLUMN 9. MACHINE. Enter the generic name for the type of punch card machine to be used for the steps entered in Column 7 (e.g., "INTERPRETER", "REPRODUCER").

COLUMN 10. PROCEDURE. Enter the processing to be applied to the steps entered in Column 7 (e.g., "INTERPRET UPPER LINE (1-60)").

DA FORM 4751-R—RECOVERY INSTRUCTIONS PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) this form was prepared or revised.

BLOCK 2. RUN ID. Enter the identification code assigned to the run. (See TB 18-103).

BLOCK 3. RUN TITLE. Enter the descriptive title assigned to the computer code.

COLUMN 4. ID CODE. When necessary to key recovery messages to a subset of run, i.e., program step, etc., enter the appropriate subset identification.

COLUMN 5. RECOVERY INSTRUCTIONS. Enter the recovery instructions.

BLOCK 6. REMARKS. Enter any clarifying information for the other entries.

DA FORM 4752-R—PROGRAM REVISION PREPARATION INSTRUCTIONS

BLOCK 1. DATE. Enter date (YYMMDD) the program was initially accepted by validation test.

BLOCK 2. PROGRAM ID. Enter identification code assigned to the program (See TB 18-103).

BLOCK 3. PROGRAM NAME. Enter the name assigned to the program.

COLUMN 4. REV NO./DATE. Changes to the program will be numbered sequentially; the date the change is implemented also will be entered.

COLUMN 5. DESCRIPTION OF REVISION. Enter a narrative description of the revision in sufficient detail to serve as documentation for a change or as a corrector description for ADS change packages as defined in AR 18-1. Include authority for change and change package number, if applicable.

SECTION 5. ADS SYSTEM AND COMPONENT IDENTIFICATION

4.22. INTRODUCTION. This section describes a means of providing for unique identification of a system or subsystem and its components (i.e., cycles, runs, programs, files, products, documents). The unique identification (ID) of each component will contain the assigned system ID which links all the components of a system. The complete system ID for all DOD designated ADS an Standard Army Multicommand Information Systems (STAMMIS) will be assigned and monitored by HQ, USACSC. For all other ADS, only the high order position (in some cases the two high order positions) will be assigned by HQ, USACSC; the remaining positions will be assigned and controlled by the ARSTAF agency or MACOM having approval authority for the ADS as shown in the table of Approval Authority Support ID. Each Agency and MACOM will ensure that no duplicate ID is assigned to ADS under their control.

4.23. SYSTEM IDENTIFICATION. The system ID is comprised of three alphabetic characters, as described below.

a. Support Identifier -- The first character, or in some instances, the first two characters of the system ID identify the Agency or MACOM which has approval authority for the system.

b. System identifiers -- The second and third characters of the system ID uniquely identify a system. If an ADS is comprised of subsystems, the characters will uniquely identify the subsystem as an ADS. NOTE: If the support identifier for an Agency or MACOM consists of two characters, then th second character of the support identifier will be the first character of the system identifier.

4.24. ADS SYSTEMS COMPONENTS IDENTIFICATION CODES. The following paragraphs describe the structure for ID codes to be assigned by the developer to the components of an ADS. Due to the wide variety of hardware and software in us within the Army, an explicit component ID structure cannot be developed here for all ADS. However, the system ID code must be an integral part of all component ID codes.

(NOTE: In the following examples, the system ID code "ABC" is used as a sample.)

a. Run ID Code -- this ID code will be used to identify a specific cycle. Examples: ABC01 or ABCM1.

POSITION	ENTRY
1, 2, 3	Must contain System ID Code
4 - 6	A three position numeric code that identifies the run uniquely within the system or subsystem.

b. Cycle ID Code -- this ID code will be used to uniquely identify a cycle. Examples: ABCD01 or ABCM01.

POSITION	ENTRY
1 - 3	Must contain System ID Code
4	An alphabetic code which describes the frequency with which a run is executed. (ref: AR 18-12-1, SDE 10-8). In addition the lett "R" will be used to describe "as required" runs.
5 - 6	A two-position numeric code that will uniquely identify the run.

c. Program/Task ID Code -- this ID code will be used to identify application programs/tasks. EXAMPLE: ABC01P. Positions 1 through 6 are u as specified below. Positions 7 and 8 are optional, but, if used, must be standardized within each system by the system developer. The program ID is also used as the basis for file identification.

POSITION	ENTRY
1 - 3	Must contain the system ID code.
4 - 5	Must be assigned and monitored by the system developer.
6	This position designates a program, and must contain one of the following codes: P = program in operational or production status. T = program in test status. D = program in development status.

d. File ID Code -- this ID code identifies files in a system. EXAMPLES: A01ABC or M01ABC.

POSITION	ENTRY
1 - 3	Must contain the System ID Code.
4 - 5	Must be identical to positions 2 and 3 of program ID where the file is initially created or accessed.
6	Any alphabetic character except "P", "S", or "U". This position is assigned and monitored by the developer of the ADS.

e. Control Card ID Code -- used to identify control cards for use in specific programs. Example: P01ABC-CNTL01.

POSITION	ENTRY
1 - 6	Must contain Program/Task ID.
7 - 11	"-CNTL")(constant entry)
12 - 13	Control Card Sequence Number. Assigned by the system developer to uniquely identify the control card within a specified program.

f. Product Control Number (PCN) -- each printed report, microform and output punched card file will have its own unique PCN. The RCS (where assigned) and the PCN must appear on every page of printed reports. EXAMPLES: ABC-999)

POSITION	ENTRY
1 - 3	Must contain the system ID code
4	Must contain a hyphen
5 - 7	Will contain a three-position numeric or alphanumeric code that sequentially identifies output products of the ADS.

g. ADS Manual (ADSM) Identifier -- the following structure will be used to identify each document type of ADS documentation.

TYPE OF PUB	PUB SERIES	ADS CODE	SYSTEM ID	MANUF CODE	DOC TYPE
ADSM	18-	L01-	AGC-	XXX-	UM

(1) Type of Publication -- This entry will always contain "ADSM" to designate an ADS manual.

(2) Publication series -- This entry will always contain "18" to designate Army automation.

(3) ADS Code -- The three-digit code assigned to the system as shown in paragraph 19.2 b, above.

(4) System identification -- Three-character identification assigned to the system IAW this section.

(5) Manufacturer Code -- Three-character code assigned to the manufacturer IAW AR 18-3.

(6) Document Tyep -- A two-position abbreviation which identifies the document type. Authorized abbreviations are shown below.

ABBREV	MEANING
FD	Functional Description
RD	Data Requirements Document
SS	System/Subsystem Specification
PS	Program Specification
DS	Data Base Specification
UM	User's Manual
OM	Computer Operation Manual
MM	Program Maintenance Manual
PT	Test Plan

TABLE 4-1. APPROVAL AUTAHORITY SUPPORT ID

IDENTIFIER	AGENCY/MACOM
A, W	US Army Computer Systems Command
BA thru BH	US Army Intelligence and Security Command
C	US Army Communications Command
D, BN thru BZ	US Army Forces Command
E	US Army Europe
FA thru FH	US Army Forces Readiness Command
FI thru FQ	Military Traffic Management Command
FR thru FZ	US Army Criminal Investigation Command
GA thru GH	The Inspector General and Auditor General
GI thru GQ	The Judge Advocate General
GR thru GZ	The Adjutant General
HA thru HH	Comptroller of the Army
HI thru HQ	Chief, Army Reserves
HR thru HZ	Chief, National Guard Bureau
J, K, L	US Army Materiel Development and Readiness Command
MA thru MH	Deputy Chief of Staff for Logistics
MI thru HQ	Deputy Chief of Staff for Operations and Plans
MR thru MZ	US Army and Air Force Exchange Service
NA thru NN	The Surgeon General
NI thru NZ	US Army Western Command
P	Eighth Army
QA thru QH	Asst Chief of Staff for Intelligence
QI thru QZ	Chief of Military History
QR thru QZ	Chief of Chaplains
R	US Army Health Services Command
SA thru SH	Deputy Chief of Staff for Research, Development, and Acquisition
T, X	US Army Training and Doctrine Command
V	US Army Japan
U	Director, Army Chief of Staff for Automation and Communication
Y, I	US Army Corps of Engineers
ZA thru ZM	Deputy Chief of Staff for Personnel

APPENDIX A

FORMS

This appendix is composed of examples of Department of the Army and Fort Benjamin Harrison forms referenced in other sections of this text. They may be used for practice and familiarization in completing the forms, or for reference to the sections where their use is described.

LOGIC CHART

For use of this form, see TB 18-111; the proponent agency is CSA.

1. DATE

2. ID

3. TITLE

DA FORM 4735-R, APR 83 REPLACES DA FORM 4735, 1 NOV 78, WHICH IS OBSOLETE.

DECISION TABLE

For use of this form, see TB 18-111; the proponent agency is DCSOPS

1. TABLE TYPE ☐ OPEN ☐ CLOSED

2. DATE

3. ID

4. TITLE

5. SEQ NO.	6. CONDITIONS/ACTIONS	7. RULES 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

FILE SPECIFICATION

For use of this form, see TB 18-111; the proponent agency is DCSOPS.

1. DATE

2. ID	3. TITLE

4. FILE DESCRIPTION

5. STORAGE MEDIA	6. RECORD FORMAT	7. FILE ORGANIZATION	8. SECLAS/PRIV
9. TAPE REC MODE	10. LABEL FORMAT	11. SEQUENCE	12. PARITY

13. TAPE BPI/TRACK	14. MIN-MAX SPACE REQ	15. BLOCK FACTOR

16. RETENTION		
a. UPON TERMINATION	b. UPON UPDATE	c. DUPLICATE(S)

17. RECORD TITLE	18. RECORD ID	19.

20. REMARKS

RECORD SPECIFICATION

For use of this form, see TB 18-111; the proponent agency is DCSOPS.

1. DATE

2. ID

3. TITLE

4. DESCRIPTION

5. LENGTH

6. SECLAS/PRIV

7. POSITION	8. FIELD	9. FIELD TITLES	10. CLASS	11. LENGTH	12. REMARKS

DA FORM [illegible] REPLACES [illegible] WHICH IS OBSOLETE.

INPUT – PROCESS – OUTPUT

For use of this form, see TB 18-111, the proponent agency is DCSOPS.

1. DATE

2. MODULE ID

3. MODULE NAME

4. VERSION NO.

5. ANALYST

6. PROGRAMER

7. INPUT	8. PROCESS	9. OUTPUT

DA FORM 4739-R, APR 83 REPLACES DA FORM 4739, NOV 78, WHICH IS OBSOLETE

INPUT–PROCESS–OUTPUT CROSS REFERENCE

For use of this form, see TB 18-111; the proponent agency is DCSOPS.

1. DATE

2. MODULE ID

3. MODULE NAME

4. VERSION NO.

5. ANALYST

6. PROGRAMER

7. IPO NUMBER			8. EXTENDED DESCRIPTION ITEMS	9. PAGE REF
a. INPUT	b. PROCESS	c. OUTPUT		

DA FORM 4740-R, APR 83 REPLACES DA FORM 4740, NOV 78, WHICH IS OBSOLETE.

INPUT–PROCESS–OUTPUT EXTENDED DESCRIPTION, PROCESS NUMBER For use of this form, see TB 18-111; the proponent agency is DCSOPS.			1. DATE
2. MODULE ID	3. MODULE NAME		4. VERSION NO.
5. ANALYST		6. PROGRAMER	
7.			

DA FORM 4741-R, APR 83 REPLACES DA FORM 4741, NOV 78, WHICH IS OBSOLETE.

CYCLE SUMMARY For use of this form, see TB 18-111; the proponent agency is DCSOPS.	1. DATE
2. CYCLE ID	3. CYCLE TITLE
4. SUMMARY	

DA FORM 4742-R, APR 83 REPLACES DA FORM 4742, NOV 78, WHICH IS OBSOLETE.

best copy available

CONTROL CARD PREPARATION For use of this form, see TB 18-111; the proponent agency is DCSOPS.	1. DATE

2. RUN ID	3. RUN TITLE
4. CONTROL CARD ID	5. CONTROL CARD TITLE

6. CONTROL CARD PURPOSE AND REMARKS

7. USER INFORMATION, IF REQUIRED

8. CONTROL CARD INSTRUCTIONS

FIELD *a.*	POSITION *b.*	PREPARATION INSTRUCTIONS *c.*	RESPONSIBLE AGENCY *d.*

DA FORM 4743-R, APR 83 REPLACES DA FORM 4743, NOV 78, WHICH IS OBSOLETE.

CONTROL CARD PREPARATION *(CONTINUATION)* For use of this form, see TB 18-111; the proponent agency is DCSOPS.	1. DATE
2. RUN ID	3. CONTROL CARD ID *(See Block 4, DA Form 4743)*

4. CONTROL CARD INSTRUCTIONS *(See Block 8, DA Form 4743)*

FIELD *a.*	POSITION *b.*	PREPARATION INSTRUCTIONS *c.*	RESPONSIBLE AGENCY *d.*

DA FORM 4743-1-R, APR 83 REPLACES DA FORM 4743-1, NOV 78, WHICH IS OBSOLETE.

RUN SETUP—CONSOLE OPERATING INFORMATION

For use of this form, see TB 18-111; the proponent agency is DCSOPS.

1. DATE

2. RUN ID

3. RUN TITLE

4. SECLAS/PRIV

5. PERIPHERAL EQUIPMENT

TAPE DRIVES ____________

DISK DRIVES ____________

CARD READER ____________

CARD PUNCH ____________

PRINTER ____________

OTHER ________ ________

6. CORE REQ

7. POINT OF CONTACT

8. RUN AS OF

9. RUN DATE

10. CONT CARD ID

11. DATE/TIME SCH IN

12. DATE/TIME SCH OUT

13. SYSTEM

14. RUN NO.

15. EST TIME

16. LANGUAGE

17. ID CODE	18. FILE ID	19. VOLUME NO.	20. I/O	21. MEDIA	22. UNIT	23. RETN	24. SECLAS/PRIV	25. SOURCE	26. DISPOSITION	27. OPERATOR NOTES

28. REMARKS

DA FORM 4744-R, APR 83

REPLACES DA FORM 4744, NOV 78, WHICH IS OBSOLETE.

RUN MESSAGES AND RESPONSES

For use of this form, see TB 18-111; the proponent agency is DCSOPS.

1. DATE

2. RUN ID

3. RUN TITLE

4. MESSAGE DISPLAYED	5. ACTION REQUIRED

6. REMARKS

DA FORM 4745-R, APR 83 REPLACES DA FORM 4745, NOV 78, WHICH IS OBSOLETE

OUTPUT REPORTS

For use of this form, see TB 18-111; the proponent agency is DCSOPS.

1. RUN ID

2. DATE

3. FILE ID	4. MEDIA	5. PCN/RCS	6. SECLAS/ PRIV	7. PRINTER					8. VOLUME	9. REPORT TITLE	10. DISPOSITION
				PAPER a.	NO. OF CYS b.	TAPE c.	PRINT POS d.	LINE UP e.			

11. REMARKS

DA FORM 4748-R, APR 83 REPLACES DA FORM 4748, NOV 78, WHICH IS OBSOLETE.

REPRODUCED OUTPUT REPORTS

For use of this form, see TB 18-111; the proponent agency is DCSOPS.

1. RUN ID

2. DATE

3. PCN/RCS	4. SECLAS/ PRIV	5. OUTPUT SIZE	6. MEDIA	7. NO. OF CYS	8. INSTRUCTIONS FOR REPORT PRODUCTION	9. DISPOSITION

10. REMARKS

DA FORM 4749-R, APR 83 REPLACES DA FORM 4749, NOV 78, WHICH IS OBSOLETE.

PCM PROCESSING INSTRUCTIONS For use of this form, see TB 18-111; the proponent agency is DCSOPS.				1. DATE
2. RUN ID	3. RUN TITLE			4. PCN/RCS
5. SOURCE		6. DISPOSITION		7. SECLAS/PRIV
8. STEP	9. MACHINE	10. PROCEDURE		

DA FORM 4750-R, APR 83 REPLACES DA FORM 4750, NOV 78, WHICH IS OBSOLETE.

RECOVERY INSTRUCTIONS For use of this form, see TB 18-111; the proponent agency is DCSOPS.	1. DATE
2. RUN ID	3. RUN TITLE
4. ID CODE	5. RECOVERY INSTRUCTIONS
6. REMARKS	

DA FORM 4751-R, APR 83 REPLACES DA FORM 4751, NOV 78, WHICH IS OBSOLETE.

PROGRAM REVISION For use of this form, see TB 18-111; the proponent agency is DCSOPS.	1. DATE

2. PROGRAM ID	3. PROGRAM NAME
4. REV NO./DATE	5. DESCRIPTION OF REVISION

DA FORM 4752-R, APR 83 REPLACES DA FORM 5752, NOV 78, WHICH IS OBSOLETE.

DOCUMENT ANALYSIS & DATA SUMMARY

_____ of _____ SHEETS

1. TYPE DOCUMENT	2. IDENTIFICATION	3. NO. OF PAGES	4. ORG CODE

5. FREQUENCY	6. VOLUME		7. NO. COPIES REC'D	8. NO. COPIES FWD
	AVG	PK		

9. SOURCE ORGANIZATION	10. DESTINATION ORGANIZATION

11. USE OF THIS DOCUMENT

12. DISPOSITION	FORWARDED ☐	SUSPENDED ☐	FILED ☐	DESTROYED ☐

13. HOW PREPARED	14. MEDIA	15. TOTAL NO. OF DATA ITEMS

16. DOCUMENTS ORIGINATING FROM THIS AND BEING UPDATED BY THIS DOCUMENT.

17. SEQUENCE BY	18. ACCESS REQUIREMENT

19. RETENTION CHARACTERISTICS

20. BACKUP DOCUMENTS THAT CAN BE USED TO RECONSTRUCT THIS DOCUMENT.

21. REMARKS

22. DATE	23. PREPARED BY

FBH FORM 32-0-2, NOV 83 REPLACES JUN 81 EDITION, WHICH IS OBSOLETE.
(ATSG-CS)

1. ITEM NO.	2. SUB NO.	3. DESCRIPTION	4. NO. OF CHAR		5. ALPHA/NUMERIC A/AN/N	6. NO. OF DEC POSITIONS
			MAX	AVG		
7. TOTAL MAX CHARACTERS OF INFORMATION →				▨	▨	▨

FBH FORM 32-0-2

GRID CHART

_____ of _____ SHEETS

1. MAJOR AREA:

2. SUB AREA:

3. ORGANIZATION:

6. Frequency →

7. Type Document →

5. End Products or Outputs (TYPE DOCUMENTS 1,2,3)

4. Input Documents (TYPE DOCUMENTS 0,1,2)

8. DATE

9. PREPARED BY

FBH FORM 32 0 14, NOV 83
(ATSG-CS)

REPLACES SEP 83 EDITION, WHICH IS OBSOLETE.

DATA ANALYSIS SUMMARY

_____ of _____ SHEETS

1a. **INPUT**

2. AREA:

3a. **OUTPUT**

← **DOCUMENT NAME** for Input & Output Data →

← TYPE DOCUMENT →

← GENERAL FLOW REFERENCE NUMBER →

← PEAK VOLUME →

← FREQUENCY →

1b.

4. **ELEMENTS OF DATA**

DATA CODE	DESCRIPTION	NO. CHAR	A/N	DEC POS

3b.

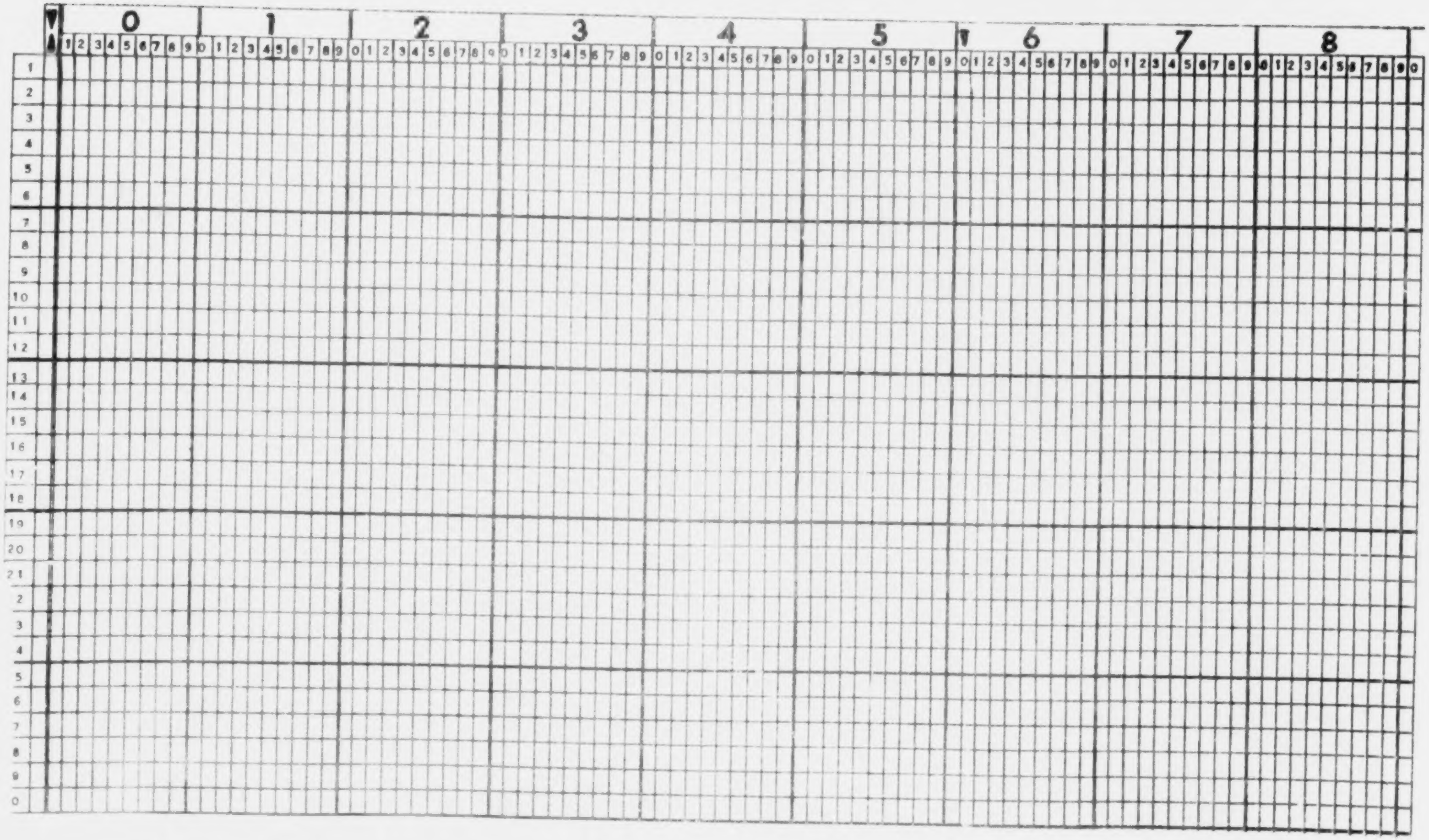
5. DATE

6. PREPARED BY

FBH FORM 32-0-17, NOV 83
(ATSG-CS)

REPLACES FBH FORM 2130, MAR 81, WHICH IS OBSOLETE.

TERMINAL SCREEN FORMAT



FORM LAYOUT

FBH FORM 7027 DEC 80

REPLACES [illegible] WHICH IS OBSOLETE

FOR INSTRUCTIONAL PURPOSES ONLY

ST 18-151

GLOSSARY OF TERMS

AMIS: Army Management Information Systems.

AMPLITUDE MODULATION: A system of radio transmission in which the carrier wave is modified by changing its amplitude (distinguished from frequency modulation).

ANALOG: Pertains to data in the form of continuously variable physical quantities. Contrast with digital...Sine Wave.

ANALYSIS: The methodical investigation of a problem and separation of a problem.

ANS COBOL: American National Standard COBOL.

ANSI: American National Standards Institute.

ARQ: Automatic Request for Repetition. A system employing an error-detecting code. So conceived that any false signal initiates a repetition of the transmission of the character incorrectly received.

ASSOCIATION: A relationship that exists between a collection of units of data based upon specified attributes.

ASYNCHRONOUS TRANSMISSION: Transmission in which information character or sometimes each word or small block is individually synchronized, usually by the use of start and stop elements. The gap between each character (word) is not of a necessarily fixed length. This type of transmission is also called "start-stop transmission".

AUDIT: The operations developed to corroborate the evidence as regards the authenticity and validity of the data introduced into the data processing problem or system.

BACKUP: The resources available to recover after degradation or failure of one or more system components (hardware or software).

BANDWIDTH: The range of frequencies available for signaling. The differences are expressed in cycles per second (Hertz) between the highest and lowest frequencies of a band.

BASIC RECORD: A group of related facts or fields of information treated as a unit to include hard copy (paper) documents, listings, and forms, as well as magnetic tape or disk records.

BATCH: Collection of all necessary data for transmission to another location or processor for processing, and then return of the results to the entry or other locations. This is usually not real time.

BATCH PROCESSING: A technique by which items to be processed must be coded and collected into groups prior to processing. A systems approach to processing where a number of similar input items are grouped together for processing during the same machine run.

BAUD: Unit of signalling speed. The speed in bauds is the number of discrete conditions or singal events per second.

BAUDOT CODE: A code for the transmission of data in which five equal-length bits represent one character. This code is used in most DC teletypewriter machines where 1 start element and 1.42 stop elements are added.

BEFORE IMAGE: A duplicate of a record, captured before that record is modified.

BENCHMARK: A point of reference from which measurements can be made.

BINDING: The firm association of the specific units of data required by a program with that program, as physically stored in a data base.

BIT: An acronym for binary digit. A single on/off signal used to construct a number or character. The smallest item of ADP information. A unit of information capable of being a storage device.

BIT MATRIX: A two-dimensional array of bit profiles for a collection of records.

BIT PROFILE: A string of bit positions, representing the presence or absence of defined attributes of a record.

BLOCK: A set of things (such as words, characters, or digits) handled as a unit. A predetermined number of records treated as a unit. Normally used for more effective computer usage.

BLOCK DIAGRAM: A diagram of a system, instrument, computer, or program, in which selected portions are represented by annotated symbols and interconnecting lines.

BLOCKING: The storage of one or more logical records in a larger physical unit of retrieval.

BLOCKING FACTOR (BF): The number of logical records contained in one physical record. The size of the physical record is based upon the blocking factor and the size of the logical record.

BPI: Bytes per inch.

BPS: Bits per second.

BROADBAND: Communications channels having a bandwidth greater than that of a voice-grade channel, and therefore capable of higher speed data transmision.

BUFFER: Area in main memory in which file is read into or written from.

BUS: A single connective link which connects multiple processing units which are co-located allowing any processing location to transmit to any other location, but only one way at a time. Two busses are usually implemented for reliability purposes.

BYTE: A sequence of adjacent binary digits (bits) operated upon as a unit. Also referred to as a character. Most common are 6-bit and 8-bit bytes.

CACHE MEMORY: That memory in a computer which contains the next most likely to be executed instruction, register extensions, stack pointers, constants, and sub-routines (common functions).

CATALOG: See INDEX.

CATHODE RAY TUBE (CRT): A device with a screen similar to that of a television set. Used to display computer-generated information.

CENTRAL PROCESSING UNIT (CPU): The unit of a computing system that includes the circuits controlling the interpretation and execution of instructions.

CHAIN: A group of records, logically related by means of links embedded in each record.

CHANNEL, VOICE-GRADE: A channel suitable for transmission of speech, digital, or analog data, or facsimile, generally with a frequency range of about 300 to 400 cycles per second.

CHARACTER CHECK: This is a control over the input of data. It is used to check any one of a number of characters in an input record. These characters can be checked for alphabetic versus numeric data, blank versus filled data, or any other data that should be consistent.

CHARTING: The technique of developing pictorial or graphic representations of some dimension of an organization or the activities of an organization.

CHECKPOINT: The record of all necessary information regarding the state of processing at some instant sufficient to permit processing to be reinitiated from that point in time.

COBOL: Common Business Oriented Language. A business data processing language.

CODE: A set of rules used to convert data from one representation to another; e.g., the American Standard for Information Interchange Code (ASCII). A number, letter, character, symbol, or any combination thereof used to represent an item of data in less space than if that item of data had been represented in its true form.

COLLATE: To compare and merge two or more similarly ordered sets of items into one set.

COMBINATION MODE: When both batch and on-line requirements exist, transactions may be accumulated and passed through the system at the same time (batch mode), while the system must be capable of handling inquiries initiated at any time.

COMPILE: To prepare machine language instructions by the use of another language. A means of converting a human type language to a machine language using the computer and a specialized conversion program.

COMPUTER JARGON: A vocabulary used by a specific group of people but not generally nor universally accepted in fields other than the one in which it originated. The jargon of the computer industry is colloquially known as "computerese."

COMPUTER OUTPUT MICROFORM (COM): Newest and potentially most economical form of output. A technique that will take output directly from a computer-generated magnetic tape, substituting a tape for a printer, and produce microform as a final output.

COMPUTER PROGRAM: A series of instructions or statements, in a form acceptable to a computer, prepared in order to achieve a certain result.

CONCENTRATOR: An electronic device which interfaces in a store and forward mode with multiple low speed communication lines at a message level and then retransmits those messages via one or more high speed communication lines to a processing site. The concentrator may be in a remote or local proximity to the processing site. The concentrator acts as a master site to the low speed lines, and as a slave to the processor.

CONDITION: An event or fact that influences the actions to be taken.

CONNECTOR: With reference to flowcharts, the means of representing the convergence of one or more flowlines into one, or the divergence of one flowline into more than one. It may also represent a break in a single flowline for continuation in another area.

CONSOLE: That part of a computer used for communication between the operator and the computer.

CONSOLE TYPEWRITER: A device tied directly to the computer to provide a two-way communication between man and machine.

CONSTANT: The quantities or messages that will be present in the machine and available as data for a program, and which will usually not change with time.

CONSTRAINT: A restriction or limit that must be met, placed upon a program or system, e.g., a limit of 48K for program size, or 3 disk drives available for use in the design of a system.

CONTENTION: (1) A method of line control in which the terminals request to transmit. If the channel in question is free, transmission begins; if the channel is not free, the terminal will have to wait until it is free. The queue of contention requests may be built up by the computer, and this can be done in a prearranged sequence, or in the sequence in which requests are made. (2) The situation that occurs when two or more run-units attempt simultaneously to access the same unit of data.

CONTROL PROGRAM: A sequence of instructions which prescribe the series of steps to be taken by a system, a computer, or any other device.

CONVERTER: (1) A device or system which changes data from analog to digital form or vice versa. (2) A unit which changes information from one format or system of expression to another. (3) A device which serves as an intermediate unit between the central processor and the input-output equipment for either of the purposes stated above.

COUNT-DATA FORMAT: This is the normal file format associated with sequential files. The records in this format are made up of an address marker, count area, and the data area. The address of the record, in terms of cylinder, head, and record is located in the count area. The data is stored in the data area.

COUNT-KEY-DATA FORMAT: This is the normal file format associated with indexed sequential and random files. The records in this format are made up of the address marker, the count area, key area, and data area. The major difference between the two formats is the "key area", which is present in this format. The "key" of the physical record located in the data area is stored in this area. The key area can range in size from 1 to 255 bytes in length.

COUPLER, ACOUSTIC: A type of modem. The acoustic coupler is used primarily in those cases where a portable terminal is desired.

CRT: Cathode Ray Tube.

CYCLE: (1) An interval of space or time during which one set of events or phenomena is completed. (2) Any set of operations that is repeated regularly in the same sequence. The operations may be subject to variations on each repetition.

CYLINDER: The recording tracks on each disk surface physically located one above the other, and pictured as a series of concentric cylinders, one per track, seen as extending vertically through all other surfaces. The surface available in a single movement of the access arm.

CYLINDER INDEX: An area on the disk used to point to the track indexes stored on the cylinders within the prime data area.

DADS: Document Analysis and Data Sheet.

DAS: Data Analysis Sheet.

DASD: Direct Access Storage Device.

DATA: Any representation such as characters or analog quantities, to which meaning might be assigned.

DATA ANALYSIS SHEET: An analytical tool used to show the relationship between documents as well as between data elements within a system.

DATA AREA: A named subdivision of the physically addressable storage space in a data base, which may contain records and sets (or parts of sets) of various types.

DATA ATTRIBUTE: A known characteristic of a data item (e.g., a numeric field, a date, etc.).

DATA BASE: The set of data or information on which operations and conclusions can be based. A data base can be composed of one or more files.

DATA BASE ADMINISTRATOR: An individual or group of individuals assigned the responsibility for the definition, organization, protection, control, and efficiency of a data base.

DATA BASE MANAGEMENT SYSTEM: A software system designed to control both the physical and logical organization of, and access to, a data base.

DATA CELL: The smallest unit of data which cannot be further subdivided, such as a magnetic bit.

DATA CELL DRIVE: A random access drive that can store millions of alphanumeric characters or digits.

DATA CHARACTERISTIC: A known characteristic of a data item (data attribute).

DATA CONCENTRATOR: A device used in data communications to multiplex numerous low-speed communications lines into a single high-speed communications line.

DATA DESCRIPTION LANGUAGE: A language used to: 1) Describe the characteristics of data items, aggregates, sets, and areas of a data base to the DBMS (See Schema). 2) Describe the characteristics of data items, aggregates, records, sets, and areas within a subset of a data base as logically viewed from within an application program (see Sub-Schema).

DATA DICTIONARY: A glossary of terms or units of data in a data base, together with information relative to their residency in aggregates and records, and their usage by one or more applications.

DATA DIRECTORY: The internal representation of the data structure, as defined by the Data Description Language, relating data terms, descriptors, and characteristics to areas of physical storage.

DATA DISTRIBUTION CHART: A matrix that provides for the recording of flow and volume by user and document or report.

DATA ELEMENT: A specific item of information appearing in a set of data.

DATA INDEPENDENCE: The concept of separating the definitions of logical and physical data from programs, so that programs are not dependent on where or how physical units of data are stored or accessed.

DATA ITEM: A single named data entity containing no substructure; the lowest level of addressable data in which data values are physically stored.

DATA MANIPULATION LANGUAGE (DML): A language of the DBMS whose statements (verbs) are used by programmers to invoke the DBMS facilities from a host language (COBOL, FORTRAN, PL/I, etc.).

DATA NAME: A name (term) given to units of data (item, record, aggregate, set) for the purpose of uniquely identifying that unit of data.

DATA PROCESING INSTALLATION (DPI): Each separate Table of Distribution and Allowances (TDA) unit, numerically designated Table of Organization and Equipment (TOE) unit, TOE OR TDA augmentation unit, contractors, or ADPE management staff, which operates ADPE at, or based on, a single geographical location.

DATA RECORD: Synonymous with Block.

DATA SET: (1) A modem that is capable of supporting high-speed synchronous transmission. (2) A named collection of records of one or more record types.

DATA USE IDENTIFIER: The name given to the use or functions of a data element in a data system.

DEBUG: To detect, locate, and remove mistakes from a routine, or malfunctions from a computer.

DECISION RULES: The programmed criteria that an on-line, real-time system uses to make operating decisions. It is important to review periodically the decision rules that are being used by a system because the nature of problems to be solved changes over time and because new situations may have arisen which were not at first anticipated.

DECISION TABLE: A table of all contingencies that are to be considered in the description of a problem, with the actions to be taken. Decision tables are sometimes used in place of flowcharts for problem description and documentation.

DEMODULATION: The process of retrieving data from a modulated carrier wave; the reverse of modulation.

DEMULTIPLEXER (DEMUX): An electronic device which takes high speed information transmissions on a single communications line from a multiplexer and separates each of the low speed messages in that transmission so that the low speed messages are output on separate lines just as if each of the low speed lines were connected from the original point where the message began to the destination point (the location of the demultiplexer).

DERIVED DATA ITEM: A data item which is not stored physically in a data base, but whose value may be computed utilizing other data items and/or constants.

DESIGN: The specification of the working relations between the parts of a system in terms of their characteristic actions.

DEVICE MEDIA CONTROL LANGUAGE (DMCL): A DBMS-supported language which permits certain physical parameters (e.g., blocking factors, device allocation) to be specified for a data base.

DIAL-UP: Establishing a connection through the use of a dial or push-button telephone and a computer through the use of common communications lines.

DIBIT: A group of two bits.

DIFFERENTIAL FILE: A file in which is recorded updated data records representing more current data than those which are stored in a read-only data base.

DIGITAL: Pertain to data in the form of digits. Contrast with "analog."

DIRECT ACCESS: A method of retrieval in which the access key directly corresponds to the disk address.

DIRECT ACCESS STORAGE DEVICE: A device in which access time is effectively independent of the location of the data (synonymous with "direct access device").

DIRECT ORGANIZATION: File organization in which each physical record in a file is assigned a unique storage address based on some predictable relationship between the key of record and the address of that record on the disk.

DIRECTORY: Synonymous with Data Directory.

DISK: A storage device on which information is recorded on the magnetized surface of a rotating disk.

DISK CYCLIC CHECK: A procedure whereby data written on a magnetic disk is compared character by character to the data that should have been written from the computer's core storage.

DISK MEMORY: A memory unit consisting of concentric recording tracks or channels formed or deposited on the surface of a flat disk.

DISK OPERATING SYSTEM: An operating system for IBM 360 installations having direct access storage devices. Also referred to as DOS.

DMA: Direct Memory Access from controllers (microprocessors) with their ability to store memory addresses and field counts independent of the CPU.

DOCUMENT: 1) A medium and the data on it for human use, e.g., a report sheet or a book. 2) By extension, any record that has permanence and that can be read by a man or a machine.

DOCUMENT ANALYSIS AND DATA SHEET: A standardized tool used by the analyst to document information contained upon media existing within the current system, or for media proposed for a new system.

DOCUMENT ORIGINATING FROM SECTION (DOFS): A part of the Document Analysis and Data Sheet in which the data pertaining to documents updated by or originating from the document being studied are recorded.

DOMAIN: Synonymous with Data Item.

DRUM STORAGE: A storage device that uses magnetic recording or a rotating cylinder. A type of addressable storage associated with some computers.

DUAL CIRCUITRY: Operations performed on the computer twice, independently, and at the same time, the results of which are compared for accuracy.

DUPLEXING: The scheme of combining a master tape with either a tape or a series of punched cards containing pure data, plus the appropriate switching codes to produce a document.

EBCDIC CODE: A 9-bit code used in data transmission, and for data representation on magnetic storage media with some types of computers.

ECHO CHECK: A system of controlling the electromechanical functions of input and output by checking points where data is transferred into or out of the system.

EDIT: Checking the correctness of data, rearrangement of data, rearrangement of data for printing, and the insertion of characters within the data.

ELEMENT: Synonymous with Data Element.

ENCRYPTION: The function of modifying a known data stream so that it will look like a random bit stream to an unauthorized observer. At the same time, an authorized observer, after a prearranged synchronization sequence, can ascertain the correct values of the original data stream.

ENTRY: Synonymous with Data Item.

ENVIRONMENT: The aggregate of surrounding influences, things, or conditions.

EXECUTION TIME: 1) The total amount of time required to complete a given command. 2) The portion of an instruction cycle during which the actual work is performed or the operation is executed.

FACSIMILE (FAX): A system for the transmission of images. The image is scanned at the transmitter, reconstructed at the receiver, and duplicated on some form of paper.

FAIL-SOFT; When a piece of equipment fails, the programs let the system fall back to a degraded mode of operation, rather than to let it fail catastrophically and give no response to its users.

FEASIBILITY STUDY: (1) Usually the initial procedures and criteria for determination of suitability, capability, and compatibility of computer systems to various firms or organizations. A preliminary systems analysis of potential cost savings and new higher level of operations, decision making, and problem solving capacity as a result of computer procurement. (2) A study in which a projection of how a proposed system might operate in a particular organization is made to provide the basis for decision to change the existing system.

FEEDBACK: The use of parts or all of the output of a machine, process, or system, as input for another phase, as when used for self-correcting purposes.

FIELD: Synonymous with Data Item.

FILE: A collection of related records treated as a unit.

FILE MANAGEMENT: The function of creating, updating, and retrieving records within a file, utilizing a self-contained, non-procedural language.

FILE PROTECTION RING: A plastic ring that fits into a groove on a tape reel. Once the tape is mounted on the tape drive, it can be read with or without the ring inserted. However, writing cannot be done without it.

FILE UPDATE: The changing of any information stored within a field, other than the record key, in an existing record in the system.

FIXED LENGTH RECORDS: Records whose lengths are fixed throughout the file at the same predetermined number of characters. Each record in the file contains the same number of fields and each field contains the same number of bytes.

FLOWCHART: A graphical representation for the definition, analysis, or solution of a problem in which symbols are used to represent operations, data flow, and equipment.

FORM LAYOUT SHEET: A document on which the systems analyst depicts how the input or output should look.

FREQUENCY DIVISION MULTIPLEX (FDM): A multiplex system in which the available transmission frequency range is divided into narrower bands, each used as a separate channel.

FREQUENCY MODULATION: A broadcasting system, relatively free from static, in which the frequency of the transmitted wave is modulated or varied in accordance with the amplitude and pitch of the signal.

FRONT-END PROCESSOR: In distributed processing systems, a front-end processor is a computer (typically a minicomputer) that is attached to and helps the host processor. When a front-end processor is used, all communications control functions are moved from the host, or central, processor to the front-end processor.

FULL-DUPLEX: Simultaneous two-way independent transmission in both directions.

GARBAGE IN-GARBAGE OUT (GIGO): A manner of stating the requirement that accurate input must be provided if correct and accurate results are to be expected.

GENERATED ELEMENT: Data that result from an arithmetic operation or from an action.

GRANDFATHER-FATHER-SON CONCEPT: In files processing, the concept of not re-using magnetic tapes containing master files until two suceeding generations of master files have been created.

GRID CHART: An analytical tool used to show the relationship between all documents collected during study of a given system.

GROUP: A named collection of data elements within a set or entry. They may vary in frequency, format, or relationship. Relationships are: subordinate, peer, superior. Also - segment, data aggregate, periodic set, detail group, element.

GROUP FACTOR: In decision tables, the result obtained by dividing the total number of rules, as determined by 2n, where n is the total number of conditions in the condition stub, by 2. This division is repeated for each remaining result of the division until the result of 1 is reached. The individual results, or factors, state: from top to bottom of the condition stub, the number of Yes (Y) and No (N) condition entries needed for each condition.

HALF-DUPLEX: A circuit designed for transmission of data in either of two directions, but not in both directions simultaneously.

HANDSHAKING: The exchange of predetermined signals for purposes of control and identification when a connection is established between two data sets.

HARD COPY: (1) Typewritten or printed characters on a page, produced at the same time information is copied or converted into machine language that is not easily read by a human. (2) A printed copy of machine output in a visually readable form; e.g., printed reports, listings, documents, summaries, etc.

HARDWARE: (1) The mechanical, magnetic, and electronic devices are components of a computer. (2) The electronic, and mechanical equipment used for processing and storage of data, consist of cabinets, racks, tubes, transistors, wires, motors, and similar items. (3) Any piece of automatic data processing equipment.

HASHING: A method of access which involves the application of an algorithm to the access key in order to develop a disk address (also referred to as randomization or key transformation).

HEAD: A device that reads or writes information in a magnetic storage medium, usually a small electromagnet used to read or write information.

HERTZ (Hz): A measure of frequency or bandwidth. The same as cycles per second.

HIERARCHY: A set of directed relationships between two or more units of data, such that some units are considered owners, while others are members. In a hierarchical structure, unlike a network, each member may have only one owner.

HOLE COUNT: The total number of holes read or punched in each column of a card at one station is compared with the total number of holes read from the same column as it passes another station. If the results of the check are in agreement, there is a high probability that the data has been transferred correctly.

HOLLERITH: A widely used system of encoding alphanumeric information onto cards. Hollerith cards are synonymous with punched cards.

HOST COMPUTER: A computer that is connected to a stored program multiplexor and which is the base or independent computer upon which the multiplexor is dependent for certain vital functions such as program read-in, etc. In an arrangement of this sort, the multiplexor could have stand-alone capability when the event the host computer is not always available.

HOST LANGUAGE: The programming language used to write applications programs.

IDE: Interim Data Element.

IDENTIFYING ELEMENT: An element of data or field name that uniquely identifies an entity to the system.

INDEX: A collection of unique descriptors (keys) and related data addresses for use in data storage and retrieval.

INDEXED SEQUENTIAL: A file that can be processed sequentially, or, by referencing indexes, in a random mode.

INFORMATION: The result of processing data; usable knowledge.

INFORMATION PROCESSING: (1) Processing data that represents information. (2) Loosely, automatic data processing.

INFORMATION STORAGE AND RETRIEVAL SYSTEM: A system that has as its main function the storage of file items for later reuse rather than modification, and most often to maintain the file items unaltered with the exception of updating and purging. Commonly referred to as ISR.

IMBEDDED FILE: A data structure in which a member type may have multiple occurrences, all logically related to one or more owners.

INPUT: (1) The data to be processed. (2) The process of transferring data from external storage to internal storage. (3) The state or sequence of states occurring on a specified input channel.

INQUIRY: A facility that permits individuals to extract a limited set of data from a data base, with the criteria for selection to be provided at run time.

INTERFACE: A common boundary between automatic data processing systems or parts of a single system.

INTERIM DATA ELEMENT: Those data names, definitions, data use identifiers, abbreviations, data items and data codes prescribed for Army-wide use and which may be formally standardized and later published as Army, DOD or Federal standards.

INTERMEDIATE RECORD (FILE): Records or files used to pass data of a temporary nature from one process to another within a function or a system. These are Code 2 documents and are more temporary in nature than Code 1 documents.

INTERNAL CONTROL SYSTEM: Programmed controls built into the system to govern the flow of computer operations.

INTERRECORD GAP (IRG): (1) The unrecorded portion between physical records on magnetic tape. (2) An interval of space or time deliberately left between recording portions of data or records. Such spacing is used to prevent errors through loss of data or overwriting, and permits stop-start tape operations.

INVERTED FILE: Key values occur multiple times within the assorted files such that entity-identifiers are associated with values of certain attributes.

ISO: Information Systems Officer.

JOURNAL: A record or log of all environmental changes relative to the data base. It may include copies of all transactions, before/after images of updated records, time and date stamps, user and terminal ID, security breaches and the like.

KEY: A field or group of fields used in the unique identification of a record within a file.

KEYPUNCH: A keyboard actuated device that punches holes in a card to represent data.

KEY TRANSFORMATION: Synonymous with Hashing.

KEY VERIFICATION: Verification of punched cards through the use of a machine similar to a keypunch. Data is verified by rekeying the data punched in the card. The machine signals an error upon detection of a difference between what was originally punched in the card, and what was rekeyed.

LANGUAGE, COMPUTER-ORIENTED: A term for a programming language requiring a low degree of translation. Such programs usually run very well on one computer, but require very extensive translation or compiling before running on another variety of computer.

LIGHT PEN: A visual display unit provides a window into the computer. It can display a series of points, a message of thousands of characters, tables, charts, graphs, and the lines and curves of drawings. The light pen which is available with the display can detect information that is displayed, and permits the operator to change the displayed information under program control.

LIMIT CHECK: Tolerance checks that provide input, housekeeping, and accounting control. They are programmed into a system to call attention to, or reject, any data which does not conform to predetermined limits.

LINEAR UNIT: A device that follows the rules of mathematical linearity, i.e., in which the change in output due to a change in input is proportional to the magnitude of the change and does not depend on the values of the other inputs. Adders, scalars, and integrating amplifiers are considered linear, whereas multipliers and function generators are often designated as nonlinear.

LINKAGE: A mechanism for relating one unit of data to another by means of associating the disk address of the target (pointed-to) record with the source (pointing) record. Links may be embedded within the data unit itself or composed of external pointer sets which relate to each data set.

LINK PATH: Synonymous with Chain.

LINK TEST: The testing of programs together, which depend upon each other for data.

LIST: A method of relating an owner record to member records of the same type, in which links pointing to each member are embedded in or associated with the owner record (as opposed to a chain).

LOGICAL DATA: That data which the application program presents to, or receives from, the data base.

LOGICAL DATA STRUCTURE: A set of relationships that exists between two or more units of logical data as viewed by the user.

LOGICAL RECORD: (1) A group of related facts or fields of information treated as a unit. It is the entity, or all the facts pertaining to that point in time, individual, etc. (2) A collection of one or more related data item values as viewed by the user.

LOGIC FLOWCHART: The most detailed of all flowcharts. Its function is to show the detailed logical or arithmetical steps involved in a computer program.

MAGNETIC DISK: A flat, circular plate with a magnetic surface on which data can be stored by means of selective magnetization of the surface.

MAGNETIC INK CHARACTER RECOGNITION: A technique, using standard symbols, used on checks to enable them to be read by machine.

MAGNETIC TAPE: (1) A tape with a magnetic surface on which data can be stored by selective polarization of portions of the surface. (2) A tape of magnetic material used as the element in some forms of magnetic cores.

MANAGEMENT INFORMATION SYSTEM ECONOMIC ANALYSIS: The document which is generated, based on all economic and operational considerations, for satisfying the functional requirements of a project. All alternatives are examined, including the existing system. This analysis may serve as the basis for the cost vs benefit analysis.

MARK-SENSE: To mark a position on a punched card by an electrically conductive pencil, for later conversion to machine punching.

MASTER FILE: (1) A file for relatively permanent information, which is usually updated periodically. (2) A main reference file of information.

MASTER INDEX: An optional index used in conjunction with cylinder index to reference tracks within the cylinder indexes.

MATCH: A data processing operation similar to a merge, except that instead of producing a sequence of items made up from the input, sequences are matched against each other on the basis of some key. The results of the match operation, normally two matching sequence files, and possibly two additional unmatched sequence files, are further processed according to some predetermined plan.

MATRIX: (1) In mathematics, a two-dimensional rectangular array of quantities. (2) In computers, a logic network in the form of an array of input leads and output leads with logic elements connected at some of their intersections. (3) By extension, an array of any number of dimensions.

MEMBERSHIP: A characteristic of data record relative to its participation within a defined set of records owned by a record of another set.

MEMORY: Generally used to refer to internal computer storage.

MESSAGE SWITCHING: The technique of receiving a message, storing it until the proper outgoing line is available, and then retransmitting. No direct connection between the incoming and outgoing lines is set up as in line switching, e.g., a telephone call.

MERGE: To combine two or more sets into one, usually in a specified order.

MICROCODE: Instructions (subordinates) in a subordinate language (machine, assembler, or compiler) operating in a control memory which executes a macro instruction (computer within a computer).

MICROCOMPUTER: A full computer on a plug-in type board utilizing a microprocessor with an 8-bit word. It is possible that the microcomputer board does not have a power supply mounted on it. This means that power must be supplied from an external source.

MICROFICHE: A sheet of microfilm containing multiple microimages in a grid pattern. It usually contains a title which can be read without magnification. The individual microimages must be read with the aid of a projection machine.

MICROFILM: A fine grain, high resolution film containing an image greatly reduced in size from the original. Used as a verb, the recording of microphotographs on film.

MICROPROCESSOR: A single chip or a chip set making up a CPU of 4 or 8 bit words.

MICROWAVE: Electromagnetic waves of extremely high frequency. Depending on their position in the band, microwaves may vary in length from 1 mm to 50 cm.

MINICOMPUTER: A computer with a 16-bit size.

MIDICOMPUTER: A computer which uses a 24- or 32-bit word size.

MISO: Management Information Systems Officer.

MODEL: A representation. To produce information, the model must be a verbal or mathematical expression describing a set of relationships in a precise manner used to predict actions or events.

MODEM: (Modulator/Demodulator) - an electronic device which converts digital signals to an analog form (frequencies) for transmission over a communication line and then converts the analog signal back to its original digital form.

MODULATION: The process by which some characters of one wave is varied in accordance with another wave or signal. This technique is used in data sets and modems to make business machine signals compatible with communications facilities.

MODULUS 10/11: Two methods of arriving at self-checking numbers.

MULTIDROP LINE: A single line or circuit interconnecting several stations.

MULTIPLEXER (MUX): An electronic device which combines information from multiple low speed communications lines for transmission on a single high speed line. The total bit capacity of the input lines cannot exceed the bit capacity of the single high speed line.

MULTIPLEXING: The division of a transmission facility into two or more channels either by splitting the frequency band transmitted by the channel into narrower bands, each of which is used to constitute a distinct channel (Frequency Division Multiplexing); or by alloting the common channel to several different information channels, one at a time (time division multiplexing).

MULTI-PORT: Use of the multi-port modem to split the communication line at the terminal end.

MULTI-PROGRAMMING: A technique for handling numerous routines or programs simultaneously by overlapping or interleaving their execution; i.e., thus permitting more than one program to time-share machine components.

MURPHY'S LAW: A theory which states that if anything conceivable can go wrong, it most probably will do so, at the most inopportune time.

NARROW BAND: Facilities capable of transmission of data at speeds of up to 300 bits per second.

NAVIGATION: The process by which a program is coded to follow access paths defined by the data base structure.

NETWORK: A set of directed relationships between owner and member records in a data base, such that a single member record may belong to one or more owner relationships. Also in communications, a network refers to the physical equipment used to connect locations for the purpose of transmitting and receiving information.

OBJECTIVE: A clear and precise statement of what is to be accomplished.

OCCURRENCE: A specific representation of a given data item, aggregate, record, or set.

OFFLINE: Pertains to equipment or devices not under the direct control of the central processing unit.

ON LINE: 1) Pertaining to equipment or devices under direct control of the central processing unit. 2) In communications; a direct connection between a terminal and a processing site which allows immediate communications between the two locations as required. The terminal may be local with a direct wire connection, or remote with a connection over communications facilities.

ON-LINE SYSTEMS: Systems which receive or transmit information as it is originated, requested, or generated. Input data enters the computer directly from the user and/or output data is transmitted directly to the user.

OPERATING SYSTEM: An integrated collection of service routines for supervising the sequencing of programs by a computer. Operating systems may perform debugging, input and output, accounting, compilation, and storage assignment tasks.

ORGANIZATION AND PERSONNEL PLAN (OPP): A document that provides Headquarters, Department of the Army with projections of changes to the organizational structure (TOE/TDA) necessary to operate the system.

OUTPUT: 1) Data that has been processed. 2) The state, or sequence of states, occuring on a specified output channel. 3) The device, or the collective set of devices, used for taking data out of a system. 4) A channel for expressing the state of a device or logic element. 5) The process of transferring data from internal storage to external storage.

OWNERSHIP: A characteristic of a data record relative to its participation within a defined set, in which records of another type exist as members subordinate to it.

PACKING DENSITY: The number of useful storage elements for each unit of dimension; e.g., the number of bits stored in each inch of magnetic tape or drum track.

PACKING FACTOR: The percentage of the reserved space on a direct access storage device that is actually being used to store records.

PAPER TAPE: A tape made of paper into which holes are punched to represent data characters.

PARITY BIT: A binary digit appended to an array of bits to make the sum of the array always an even or odd number.

PARITY CHECK: A check that tests whether the number of ones (or zeros) in an array of binary digits is odd or even.

PERIPHERAL EQUIPMENT: The auxiliary machines that may be placed under the control of a central computer. Peripheral equipment may be used either on-line or off-line depending upon computer design, job requirements and economics.

PHASE MODULATION: Radio transmission in which the carrier wave is modulated by changing its phase to transmit the amplitude and pitch of the signal.

PHYSICAL DATA: That data which the DBMS stores on, or retrieves from, a storage medium.

PHYSICAL RECORD: One or more logical records contained within a block. The terms block and physical record are synoymous. The single unit of data accessed by the hardware from some storage medium.

PHYSICAL STRUCTURE: A set of relationshps that exist between two or more units of physical data.

POINTER ARRAY: A data record which participates in set relationship as an owner; and which, within it, the disk addresses of all members of the set. (See List).

POLARIZATION: A positive or negative condition.

POLLING: A means of controlling communications lines. The communication control device sends signals to a terminal saying, "Terminal A; have you anything to send?" -- if not -- "Terminal B; have you anything to send?" -- and so on. Polling is an alternative to contention. It makes sure that no terminal is kept waiting for a long time.

PRIME AREA: The area on a direct access device that contains the greatest part of a file.

PRIME KEY: An access key which uniquely identifies a record.

PRINTER: An output device that spells out computer results as numbers, words, or symbols on hard copy.

PRIVACY: The techniques used for limiting and/or preventing access to system information based on the content of that information.

PROBABILITY: In statistics, the relative frequency of an occurrence of an event as measured by the ratio of the number of cases or alternatives favorable to the event to the total number of cases or alternatives.

PROGRAMMER: One who prepares programs for a computer.

PROTOTYPE TEST: The initial operations experience during which the capability of an automated system to accomplish predetermined processing requirements is measured and evaluated, using file records or supplementary data obtained from actual operations over an extended period of time. The prototype test is not analogous to the prototypes defined in AR 11-25 and DA Pam 11-25.

PUBLIC SWITCHED NETWORK: Any switching system that provides circuit switching to many customers. In the USA there are four such networks; TELEX, TWX, telephone, and broadband exchange.

PUNCHED CARD: A card punched with holes to represent data in the form of numbers, letters, or special characters.

QUALIFIER: A relationship or value criterion used to select one or more logical records.

QUANTITATIVE ANALYSIS: The analysis of a substance in order to determine the amounts and proportions of its elements.

QUANTITATIVE ELEMENTS: Elements of data or fields used in an arithmetic operation or measure.

QUESTIONNAIRE: A form that contains a number of questions for which an analyst needs facts or opinions that may contribute to the success of a systems study.

QUEUE: A line or group of items waiting for the attention of the processor. Usually in core and chained together by address words.

RANDOM ACCESS: 1) Pertains to the process of obtaining data from, or placing data into, storage when the time required for such access is independent of the data most recently placed into, or retrieved from, storage. 2) Pertains to a storage device in which the access time is effectively independent of the location of the data.

RANDOMIZING: Synonymous with Hashing.

REALM: Synonymous with Data Area.

REAL TIME: 1) Pertaining to the performance of a calculation during the actual time that the related physical process transpires so that results of the calculation can be used in guiding the physical process. 2) The entry of information into a network via a terminal and the immediate processing of the task.

RECOVERY: 1) The necessary actions required to bring a system to a predetermined level of operation after a failure has occurred. 2) The process of reconstituting a data base to its correct and current state following a partial or complete hardware, software, operational, or processing error.

RECORD: 1) A collection of related items of data, treated as a unit. 2) To enter data on hard copy documents to be used as input into a data processing system, or to enter data into the files of the data processing system.

REDUNDANCY: 1) In the transmission of information, redundancy is the fraction of the gross information content of a message that can be eliminated without the loss of any of the essential information. Expressed numerically, it is one minus the ratio of the net information content to the gross information content, expressed in percent. 2) A situation where there are multiple occurrences of a particular unit of data in a data base.

RELATION: A file which contains tuples (records) of a certain type.

RELATION MODEL: A data structuring approach first proposed by E. F. Codd.

RELATIONSHIP: The definition of a single association between two units of data.

REMOTE DATA CONCENTRATION (RDC): Programmable minicomputers that can be used in place of, or in addition to, modems or multiplexors at the terminal end.

REORGANIZATION: The process of physically reorganizing the placement of records in a data base.

REQUESTOR: An individual desiring to use the data in a data base.

RESPONSE TIME: The time measured from the depression of the "ENTER" or similar key at a terminal, to the display of the first character on either the CRT or printer at that same terminal location, regardless of the total path required for the complete transaction.

RESTRUCTURE: The process of adding, deleting, or rearranging units of data in a data base.

RETRIEVAL: Recovery of data from storage or collection points for the purpose of obtaining information or data.

REVERSE ARITHMETIC: A method by which a computer verifies a calculation. Under this method, a calculation of A times B equals T is checked by the performance of calculating A times B equals X, then subtracting X from T, and ascertaining that the end result of the subtraction is zero.

RING STRUCTURE: A method of relating units of data (records) by means of links embedded in each record or external to, but related to each record. The link in the last member record points back to the owner record.

ROUTING: The assignment of a communications path for the delivery of a message.

ROUTING INDICATOR: An address or group of characters in a message header that defines the final circuit or terminal.

RUN: Computer operation through one complete program performance.

SAMPLING: Obtaining a value of a variable at regular or intermittent intervals.

SATURATION: The condition existing when more time is required for computer operations than is available. AR 18-1 considers that, for planning purposes, computers are saturated when operations require more than two shifts per day, five days per week (80 hours).

SCHEMA: The complete internal representation of a data base -- includes all named entities; sets, groups, data items, areas, sort sequences, access keys, and security locks.

SCIENTIFIC METHOD: Defining the problem, analysis of the problem, selecting alternative solution(s), testing of the selected solution(s), and implementation.

SCOPE: The operation, subsystem, or functional area that will be affected by a systems analysis study.

SDE: Standard Data Element.

SEARCH STRATEGY: Synonymous with Access Strategy.

SECURITY: The concept that access to units of data in a data base be limited to users with proper authority and a "need to know". Also, the techniques used for limiting physical access to information and/or encryption of data.

SELF-CHECKING NUMBER: A system of adding a check digit to identification numbers.

SEQUENCE CHECK: 1) A system of printing or stamping serial numbers on each document, for subsequent input or output control and accountability. 2) A manual or mechanized operation performed to check the sequence of records in a file.

SEQUENTIAL ACCESS: Searching in sequence through all locations until the location of the desired data is found. Magnetic tape is a good example of this type of access.

SEQUENTIAL ORGANIZATION: The organization of a sequential disk file, whereby the file is maintained sequentially based the selected key.

SEQUENTIAL PROCESSING: A method of storing or retrieving units of data in an ordered series upon reference to a single access key.

SERIAL: Pertaining to time-sequential transmission, storage of, or logical operations on, parts of a word, using the same facilities for successive parts.

SERIAL PROCESSING: A method of storing or retrieving units of data in the sequence of occurrence without regard to any ordering on access keys.

SET: A named relationship between two or more record types, in which one record type participates as the owner of the set, and one or more records of other types participates as members.

SIMPLEX CIRCUIT: A circuit capable of one-way operation only.

SIMULATE: To represent the functioning of one system by another. E.g., to represent one computer by another; to represent a physical system by executing a computer program; or to represent a biological system by a mathematical model.

SINE WAVE: An analog signal that can represent 0 and 1 conditions.

SOFTWARE: The collection of programs and routines associated with a computer; e.g., compilers, library routines, utility programs.

SORT: To arrange data or items in an ordered sequence by applying specific rules.

SOURCE DATA AUTOMATION (SDA): The basic principle of capturing information in a usable medium, at the point of origin, for further processing.

SOURCE DOCUMENT: A document originally used by a data processing system and which supplies the basic data to be input to the data processing system.

SPECIAL CHARACTER: In a character set, any character that is neither a numeral, a letter, or a blank; i.e., "/", "*", "$", "=", etc.

STANDARD DATA ELEMENT: Developed and approved data names, definitions, data use identifiers, abbreviations, data names and codes.

STATISTICAL MULTIPLEXER: A multiplexing device (sometimes called an intelligent multiplexer) which can accept more low speed lines than a standard multiplexer because of the statistical probability that all the low speed lines will not be busy at the same time.

STORED PROGRAM COMPUTER: A digital computer that, under control of internally stored instructions, can combine, alter, and store instructions as though they were data and later carry out those instructions.

SUBROUTINE: A routine that can be part of another routine.

SUB-SCHEMA: A subset of the schema, representing a portion of the data base as it appears to a user/application program. Identifying names may differ in a manner consistent with the schema.

SUPERVISOR: The computer routine, continually residing in main memory which controls internal scheduling, set up, and execution of all programs entering the computer.

SYMBOL: A representation of something by reason of relationship, association, or convention.

SYNCHRONOUS: Having a constant time interval between successive bits, characters, or events. The term implies that all equipment in the system is in step.

SYNCHRONOUS TRANSMISSION: A transmission process such that between any two significant instants there is always an integral number of unit intervals.

SYNONYM: Two or more keys which will generate the same address.

SYSTEM: 1) An organized collection of parts united by regulated interaction. 2) An organized collection of men, machines, and methods required to accomplish a specific objective.

SYSTEM APPROACH: Application of scientific methods to the study of any system.

SYSTEMS ANALYSIS: Application of scientific methods to the solution of business problems.

TABLE: Synonymous with Index.

TAPE LABEL: External or internal information used for the manual or mechanical control or accountability of tape files.

TAPE SPEED: The speed at which tapes are transported from feed to take-up reels during normal recording or reproduction.

TELECOMMUNICATION: The transmission or reception of signals, writing, sounds or intelligence of any nature by wire, radio, light beam, or any other electromagnetic means.

TELECOMMUNICATIONS CONTROL PROGRAM: The software interface which oversees operation of the telecommunication system.

TELECOMMUNICATIONS CONTROL UNIT: The control unit or hardware interface between the central processing unit and the terminal.

TELEPROCESSING: A term describing systems that transmit data from one point to another in the course of processing.

TELEPROCESSING SYSTEM: The main task of this system is to interconnect, economically and reliably remote terminals to a host computer which may perform data processing functions and/or routing of messages between terminals.

TERMINAL: A point at which information can enter or leave a communication network. An input/output device designed to receive data in an environment associated with the job to be performed and capable of transmitting entries to and obtaining input from the system of which it is a part.

TEST DECK: A deck of data cards used in testing a program for accuracy in processing, and the program's ability to detect and handle all possible types of errors.

THROUGHPUT: The productivity based on all facets of an operation; e.g., a computer with a capability for simultaneous operations of read/write/compute would have a high throughput rating.

TIMESHARING: The ability of multiple users to share the same resources in time at the task level.

TIME DIVISION MULTIPLEXING: A transmission method by which a communication channel is established between selected terminal equipment and a host processor by means of one common channel; generally at regular intervals, and by means of automatic distribution.

TIMING CHECK: A signal sent to a printer to make print wheels advance to the desired position. The amount of actual advance is sensed, converted to the printing condition the position represents, and compared with the original signal.

TOLERANCE CHECK: Limit checks that provide input, housekeeping, and accounting control. Limit checks are programmed into a system to call attention to any data exceeding or less than predetermined limits.

TOUCH-TONE: AT&T push-button telephone dialing system.

TRACK: The portion of a moving storage medium, such as a drum or disk, that is accessible to a given reading head position.

TRACK INDEX: In a direct access storage device, the lowest level of index. A track index is always created by the system and is always located on the first track of each cylinder within the prime area.

TRAILER LABEL: The last record of a tape file, which contains control totals for the file.

TREE: Synonymous with Hierarchy.

TRANSACTION: A named collection of one or more data items representing a unique event or occurrence. The event may involve input, output, or a request for processing.

TRANSACTION FILE: Transactions accumulated as a batch for processing against the master file.

TRANSFER RATE: The speed at which data may be read from or written to a device, from the lowest to the highest speed and density available.

TROUBLESHOOT: Synonymous with Debug.

TUPLE: An ordered collection of one or more domains (data items) which constitute an occurrence of a record.

TURNAROUND DOCUMENT: A variation of a pre-punched card. Turnaround documents are machine generated outputs that serve, either independently or in conjunction with some other document, to gather information. This output document is sent to its destination and then returned with additional data posted to it. When the document is returned, most, if not all, necssary control information is already punched into it.

UNATTENDED OPERATIONS: The automatic feature of a station's operation which permits the transmission and reception of messages on an unattended basis.

UNBLOCKED: The condition in which one physical record contains only one logical record.

UNIT OF DATA: A named collection of logical and physical data. Unit of data are: data item, data aggregate, record, file, data area.

UNIT TESTING: Testing and debugging individual programs.

VALIDATE: To determine the accuracy and acceptability of the content of data elements such as codes.

VALIDITY CHECK: A control over input data. If there is a limited number of valid codes, the code being input may be checked to ensure that it is one of the valid codes.

VALIDITY OF DATA: The status of information after being handled by a hardware module or communications equipment. The goal of validity of data is the moving of data without error.

VARIABLE LENGTH RECORD: Record lengths vary throughout the entire file. The length is determined by the number of actual data positions needed for each individual record.

VERIFY: (1) To determine whether data has been transcribed accurately. (2) To check the results of keypunching.

VOICEBAND: Facilities capable of data transmission from 300 to 9600 bits per second.

VOICE FREQUENCY/TELEPHONE FREQUENCY: Any frequency within that part of the audio-frequency range essential for the transmission of speech or commercial quality in the range from 300 - 3400 cps.

WAIT: The condition a real-time program meets when it requires information from a file-storage unit and is forced to "wait" until the unit accesses the required file record and brings it into main memory. File-oriented systems have this characteristic that leads to multiprogrammed approaches by interleaving and overlapping "wait" times for one program to achieve process time for another program.

WIDE AREA TELEPHONE SERVICE (WATS): A leased or measured dial-up service which allows: 1) A subscriber to make a predetermined number of calls for a predetermined period of time to specific geographic areas; 2) Any other user to make calls to a central site at no cost to that user. Separate charges are tabulated for intrastate and interstate WATS service. The current monthly interstate limitations for a 30-day period, before overtime charges are incurred are: 240 hours; 14,400 calls. The United States is divided into six WATS areas for control purposes.

WITH KEY: Synonymous with COUNT-DATA FORMAT.

WITHOUT KEY: Synonymous with COUNT-KEY-DATA FORMAT.

WORD: One or more contiguous bytes which, in most cases, identify the parallel processing capability of the machine to which it relates.

WRITE: 1) To transfer information to an output medium; to copy, usually from internal storage to external storage; to record information in a register, location, or other storage device or medium. 2) To transfer information, usually from main storage, to an output device.

ZERO BALANCING: A technique associated with batch controls which serves as a control over input. Each detail item is accumulated and the final total can be subtracted from the batch total. The result is a zero balance if both are correct.

ZERO SUPPRESSION: The elimination of nonsignificant zeros in a numeric field.

TESTING MATRICES

DEVELOPMENT PHASE TESTING

		Test Plan	Test Environment	Test Data	System Document and Training Package	Test Evaluation	Test Report	Certification
SDT	Module/ Program	N/A	Local Test Bed	ASD Generated or BMTF	Optional	Module/ Program Verification and Validation	N/A	N/A
	Cycle/ System	IAW TB 18-111		Benchmark Test Files (BMTF[1])	Draft	Cycle/System Verification and Validation	IAW TB 18-111	Sign-Off Required
SQT		IAW TB 18-111	Target Hardware	Live Files[1]	System Documentation in Final Draft	Qualification Verification and Validation	IAW TB 18-111	Sign-Off Required
SAT		IAW TB 18-111	Operational Environment	Onsite Live Data[1]	Final	Acceptance Verification and Validation	IAW TB 18-111	Sign-Off Required

[1] Or as specified in test plan.

OPERATIONAL PHASE TESTING
SOFTWARE CHANGE PACKAGE (SCP)

		Test Plan	Test Environment	Test Data	System Documentation	Test Evaluation	Test Report	Certification
SDT	Module/ Program	N/A	Local Test Bed	ASD Generated or Conditioned BMTF	Optional	Module/ Program Verification and Validation	N/A	N/A
	Cycle/ System	IAW TB 18-111		Conditioned BMTF[1]	Draft	Cycle/System Verification and Validation	IAW TB 18-111	Sign-Off Required
SAT		IAW TB 18-111	Operational Environment	Onsite Live Data[1]	Final	Acceptance Verification and Validation	IAW TB 18-111	Sign-Off Required

[1] Or as specified in test plan.

U.S. GOVERNMENT PRINTING OFFICE: 1986 - 641-003/20450